

R8C/35C グループ

R01AN3378JJ0100

I²C バスシングルマスタ制御プログラム(マスタ送信/受信)

Rev.1.00

2017.05.11

リスタートコンディションを発行する I²C 通信

目次

1. 要約.....	3
2. はじめに.....	3
3. 応用例の説明.....	4
3.1 プログラムの概要.....	4
3.1.1 使用周辺機能.....	10
3.1.2 使用上の注意事項.....	10
3.2 使用メモリ.....	10
4. ソフトウェア説明.....	11
4.1 使用変数.....	11
4.2 関数表.....	13
4.3 メイン処理.....	23
4.4 システムクロック設定処理.....	25
4.5 I/O ポートの初期設定処理.....	26
4.6 I ² C 初期設定処理.....	27
4.7 タイマ RA 初期設定処理.....	28
4.8 カラーセンサの初期化処理.....	29
4.9 通信パラメータのコピー処理.....	30
4.10 I ² C 通信開始処理.....	30
4.11 スタートコンディション発行処理.....	32
4.12 ストップコンディション発行処理.....	32
4.13 カラーセンサのリセット処理.....	33
4.14 光強度の計測完了待ち処理.....	35
4.15 タイマ RA 動作開始処理.....	35
4.16 測定データの読み出し処理.....	36
4.17 測定結果の処理.....	38
4.18 通信再開待ち処理.....	39
4.19 I ² C 通信エラー処理.....	39
4.20 I ² C バスインタフェース割り込み処理.....	40
4.21 データ送信処理.....	41
4.22 連続データ送信処理.....	41
4.23 受信設定処理.....	42
4.24 データ受信処理.....	43
4.25 通信エラー終了処理.....	44
4.26 タイマ RA 割り込み処理.....	44

5. サンプルコード	45
6. 参考ドキュメント	45

1. 要約

この資料は、R8C/35C グループの I²C バスインタフェースを使用して、リスタートコンディションの発行に対応した I²C バスシングルマスタ制御プログラム(マスタ送信/受信)について説明しています。

本サンプルコードでは、一例として I²C バスインタフェースに対応したデジタルカラーセンサ(S11059-02 DT)を制御します。

2. はじめに

この資料で説明する応用例は、次のマイコン、条件での利用に適用されます。

マイコン : R8C/35C グループ

高速オンチップオシレータ : 20MHz

本アプリケーションノートは、上記グループと同様の SFR(周辺機能制御レジスタ) を持つ R8C ファミリマイコンでも使用できます。ただし、一部の機能を変更している場合がありますのでユーザーズマニュアルで確認してください。また、本アプリケーションノートで説明しているプログラムを使用される場合は十分な評価を行ってください。

3. 応用例の説明

3.1 プログラムの概要

デジタルカラーセンサ(S11059-02DT)の通信フォーマットに従うように、マスタ送信でスレーブアドレスおよび8バイトのデータを送信し、マスタ受信でスレーブアドレスの送信後に8バイトのデータを受信します。マスタ送信とマスタ受信を、交互に繰り返します。

本サンプルコードは、リスタートコンディションの発行に対応しています。マスタ送信時、5バイト目のデータ送信後にストップコンディションを発行せず、リスタートコンディションを発行して6バイト目を送信します。また、8バイト目のデータ送信後にストップコンディションを発行せず、リスタートコンディションを発行してマスタ受信モードへの切り替えを行います。

リスタートコンディションの発行を行うことで、接続先デバイスの通信フォーマットに一致するようにデータを送信すること、送信終了後に接続先デバイスからデータを受信するためにデータ転送方向を切り替えることを可能にしています。

下記使用条件において I²C バスの通信プロトコルに準拠しています。

<使用条件>

- スレーブアドレス : 7ビット
- 転送レート : 約 357kHz (Standard-mode、Fast-mode 対応)
- 転送データ長 : 1~255 バイト(スレーブアドレス含まず)
- シングルマスタ通信(マルチマスタ未対応)

図 3.1に通信フォーマットを、図 3.2にブロック図を、図 3.3および図 3.4に概略フローチャートを、図 3.5～図 3.7にタイミング図を示します。

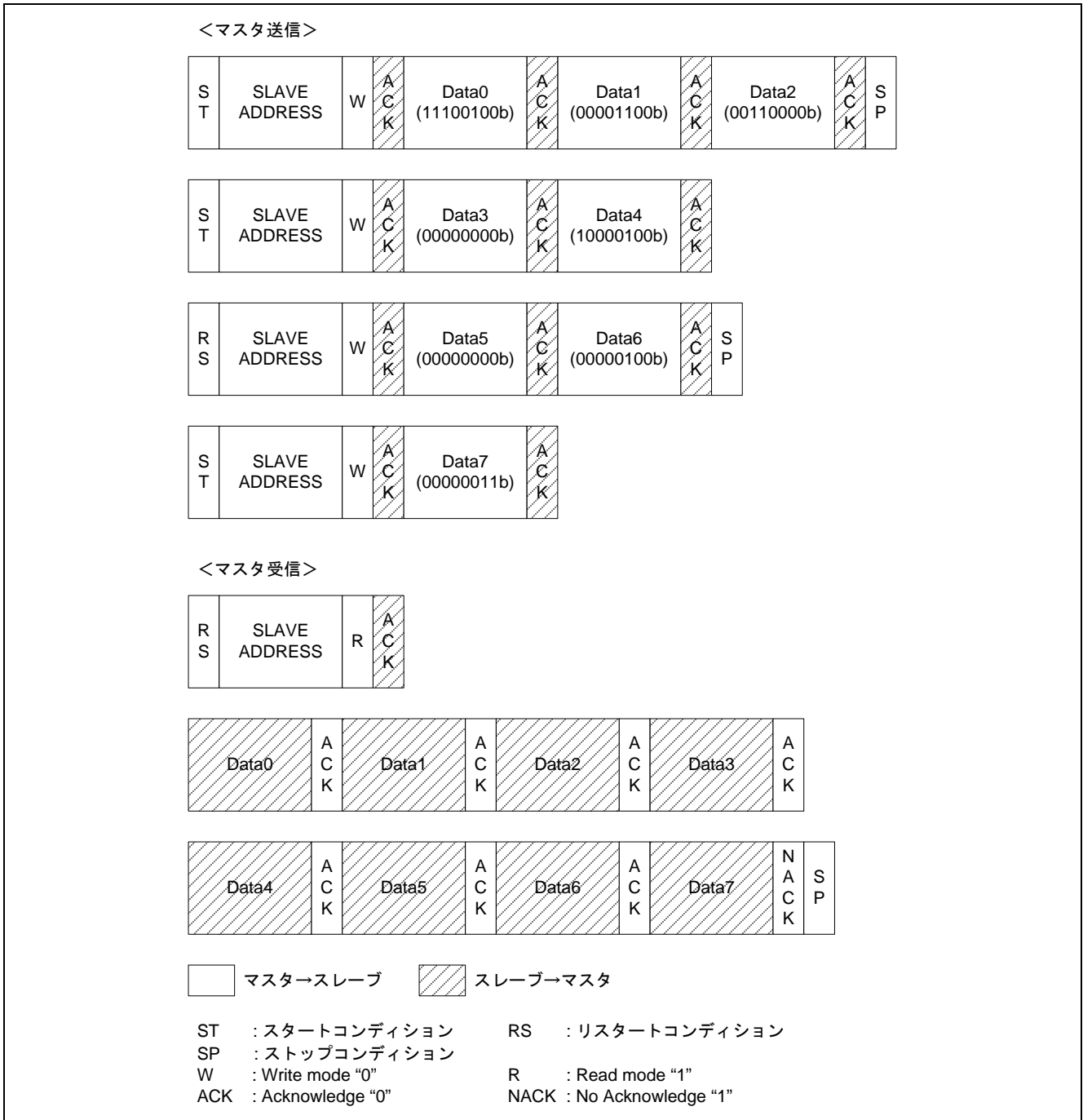


図 3.1 通信フォーマット

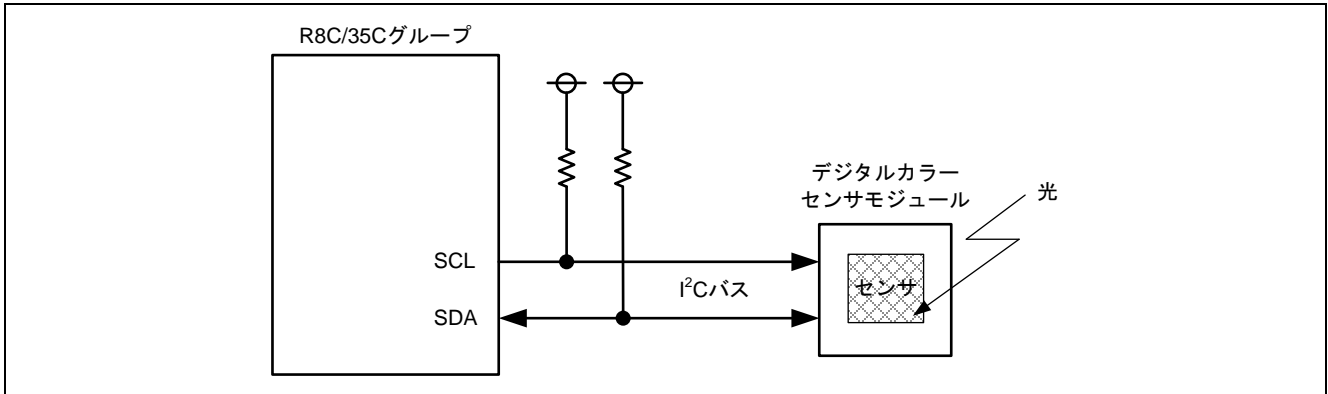


図 3.2 ブロック図

図 3.3および図 3.4中の番号は、図 3.5～図 3.7のタイミング図中のプログラムによる処理の番号に対応しています。

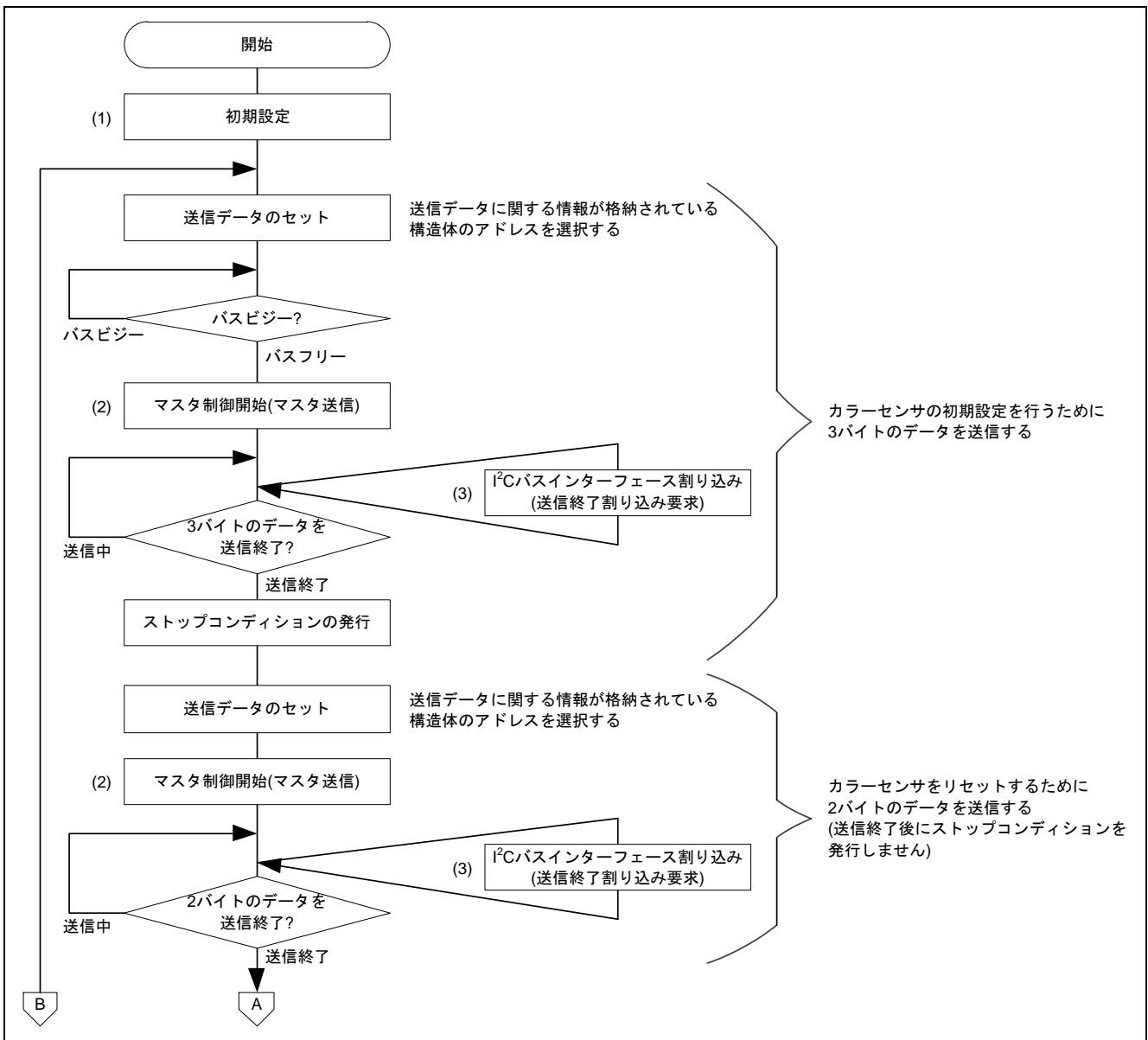


図 3.3 概略フローチャート (1/2)

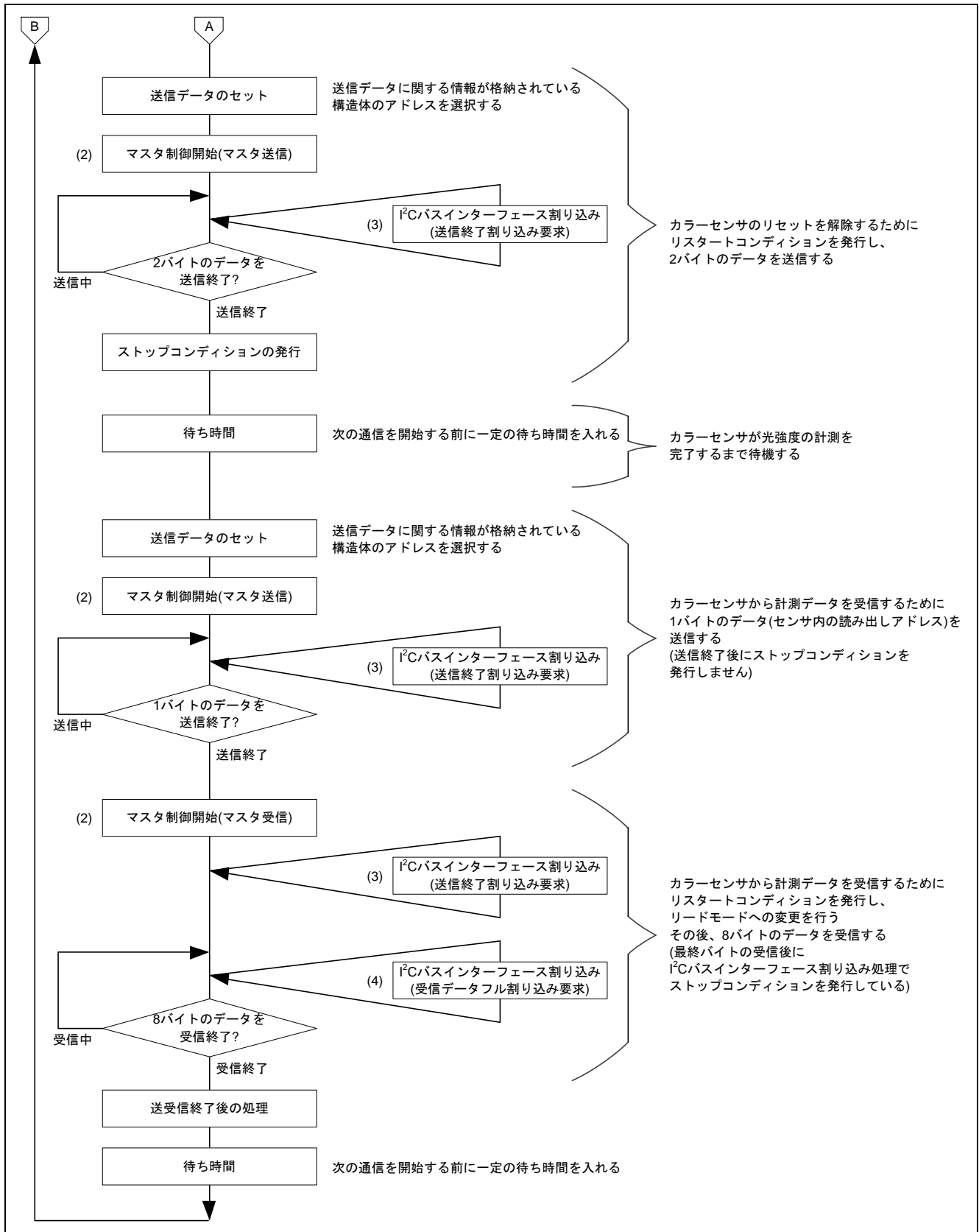


図 3.4 概略フローチャート (2/2)

処理の概要を以下に説明します。

(1) 初期設定

システムクロックおよび I²C バスインタフェース関連 SFR の初期設定と、使用する変数の初期化を行います。

(2) マスタ制御開始

スタートコンディションを発行します。

I²C バスインタフェース割り込み(送信終了割り込み要求)を許可に設定し、スレーブアドレスを送信します。

(3) I²C バスインタフェース割り込み(送信終了割り込み要求)

SCL クロックの 9 ビット目の立ち上がりで、割り込みが発生します。

<マスタ送信時>

- ・ ACK/NACK の判定をして、次バイトの送信データを設定します。

<マスタ受信時>

- ・ 送信終了割り込み要求を禁止に設定し、受信データフル割り込み要求を許可に設定します。

(4) I²C バスインタフェース割り込み(受信データフル割り込み要求)

マスタ受信の時、SCL クロックの 9 ビット目の立ち上がりで、割り込みが発生します。

次バイトの ACK/NACK を設定し、受信データを読み出します。

通信を終了するとき、ストップコンディションを発行し、受信データフル割り込み要求を禁止します。

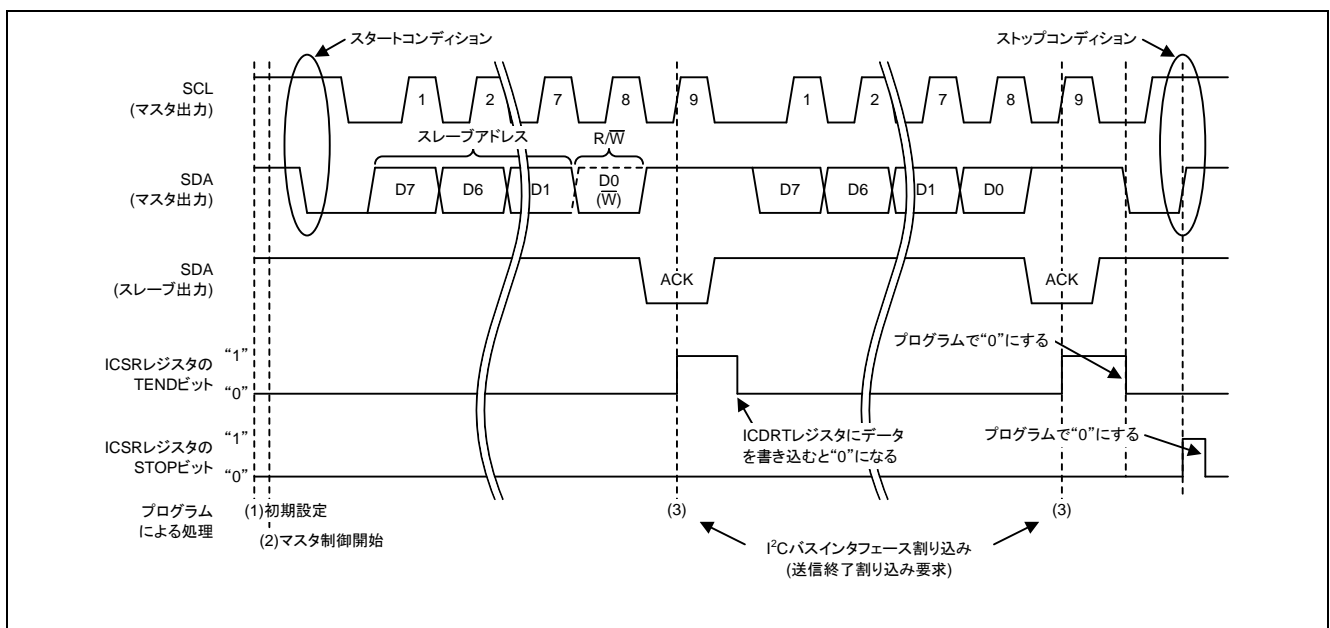


図 3.5 マスタ送信のタイミング図

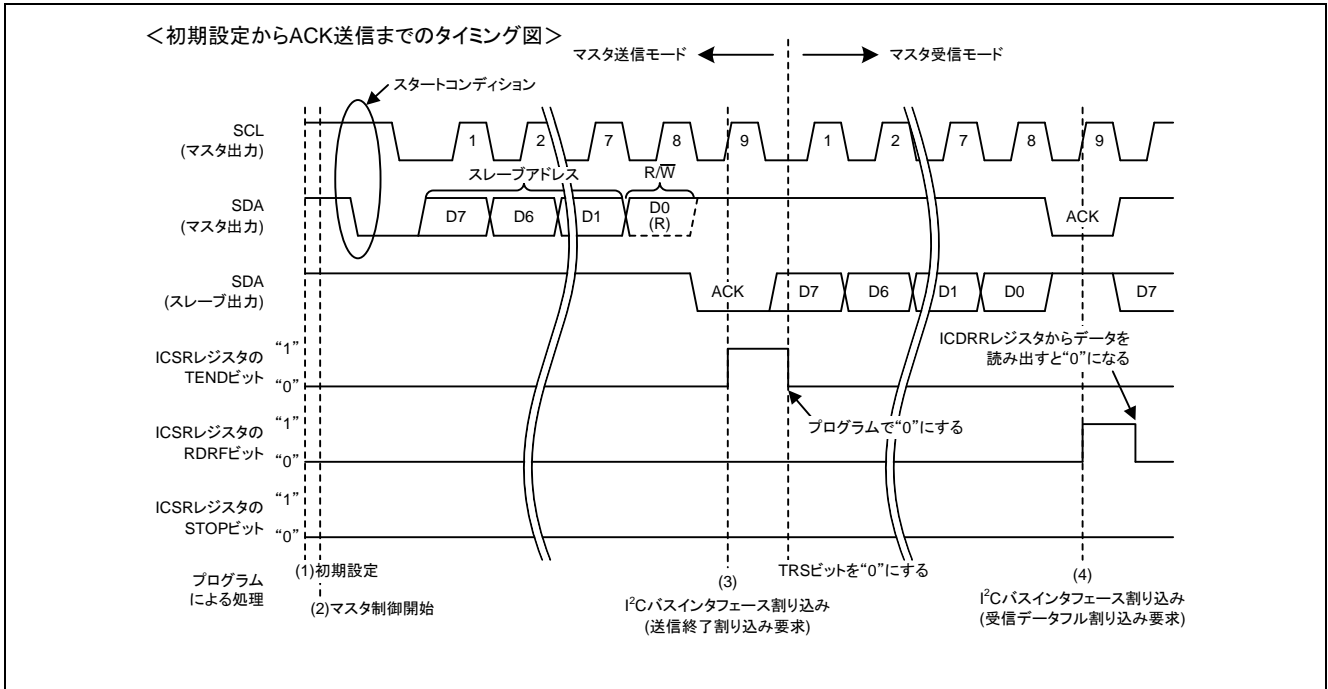


図 3.6 マスタ受信のタイミング図 (1/2)

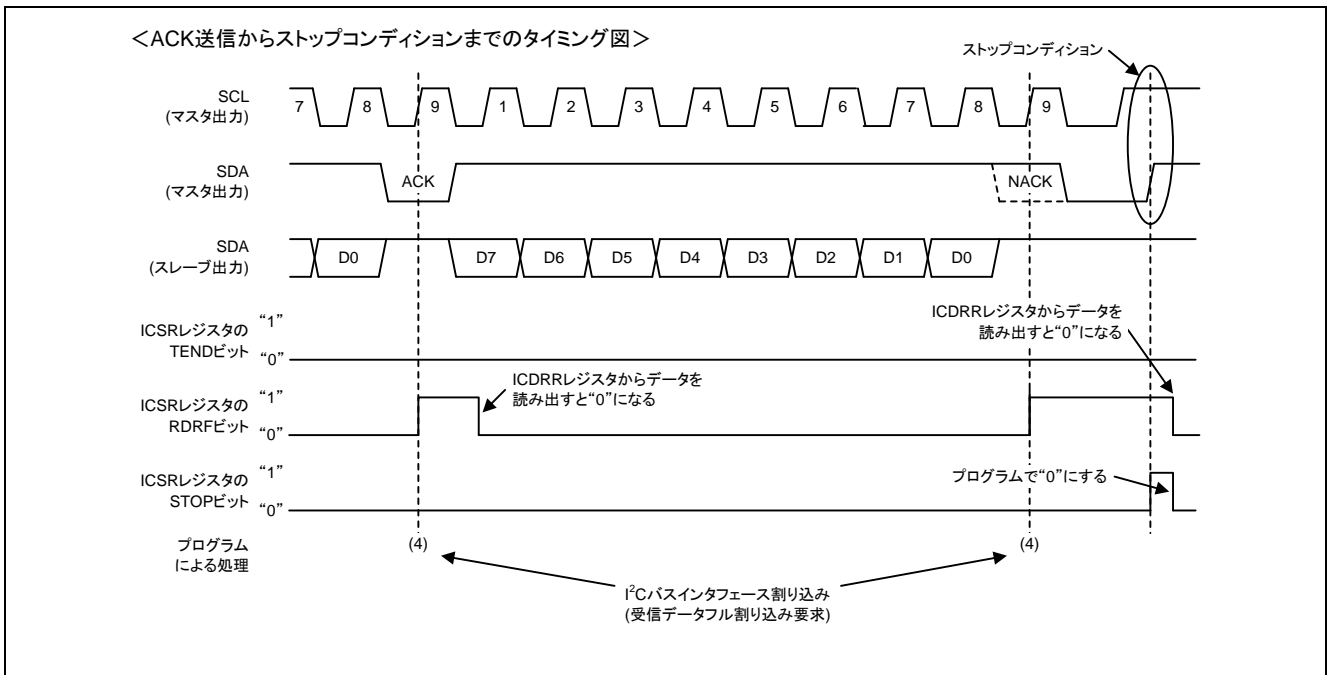


図 3.7 マスタ受信のタイミング図 (2/2)

3.1.1 使用周辺機能

I²C バスインタフェースの I²C バスインターフェースモードを以下の設定条件で使用します。

<設定条件>

- I²C バスフォーマットを使用します。
- 転送クロックは、 $f1/56$ を使用します(転送レートは、約 357kHz に設定します)。
- ウェイトなし(データとアクノリッジを連続して転送)を設定します。
- 転送フォーマットは、MSB ファーストを使用します。
- SDA デジタル遅延値は、 $3 \times f1$ サイクルを使用します。
- アクノリッジの判定は、受信アクノリッジビット(ACKBR ビット)を使用します。
- 受信データフル割り込み要求を使用します。
- 送信終了割り込み要求を使用します。
- ストップコンディション検出割り込み要求は使用しません。
- 送信データエンブティ割り込み要求は使用しません。
- NACK 受信割り込み要求およびアービトレーションロスト/オーバランエラー割り込み要求は使用しません。

<転送レートの計算式>

$$\begin{aligned} \text{転送レート} &= \text{ICCR1 レジスタの CKS0} \sim \text{CKS3 ビットによる設定} \\ &= 20\text{MHz}(f1) \div 56 \\ &\approx 357.142\text{kHz} \end{aligned}$$

表 3.1 使用端子と機能

端子名	入出力	機能
P3_5/SCL	入出力	I ² C バスのクロック入出力
P3_7/SDA	入出力	I ² C バスのデータ入出力

3.1.2 使用上の注意事項

応用例のプログラムを使用する場合の注意事項を以下に示します。

- 多重割り込みは使用しないでください。
- システムクロックに高速オンチップオシレータ(20MHz)以外を使用する場合は、「3.1.1 使用周辺機能」の転送レートの計算式を参照して、CKS0～CKS3 ビットの設定値を変更してください。

3.2 使用メモリ

表 3.2 使用メモリ

使用メモリ	サイズ	備考
ROM	2146 バイト	—
RAM	317 バイト	—
最大使用ユーザスタック	37 バイト	—
最大使用割り込みスタック	25 バイト	—

使用メモリサイズは C コンパイラのバージョンやコンパイルオプションによって異なります。上記は次の条件の場合です。

C コンパイラ : M16C Series, R8C Family Compiler V.5.45 Release 01

コンパイルオプション : -c -finfo -dir "\$(CONFIGDIR)" -R8C

4. ソフトウェア説明

応用例を実現するためのプログラム例を示します。各レジスタの詳細は「R8C/35C グループ ユーザーズ マニュアル ハードウェア編」を参照してください。

4.1 使用変数

表 4.1 定義ファイル名 : r01an3378_src.c (1/4)

変数名	サイズ	使用内容	
iic_stat_fcb iic_comstat	全 5 バイト	通信ステータス格納用構造体	
構造体のメンバ	unsigned char iic_status	1 バイト	通信ステータス
		b0	通信方向フラグ 0 : マスタ送信 1 : マスタ受信
		b1	アドレス不一致フラグ 0 : アドレス不一致なし 1 : アドレス不一致あり
		b2~b3	コマンド・エラー時のみ使用(未定義)
		b4	エラーフラグ 0 : 正常終了 1 : エラー
		b5	不使用(未定義)
		b6	コマンド要求時のみ使用(未定義)
		b7	通信中フラグ 0 : 通信終了 1 : 通信中
	unsigned char iic_slave_addr	1 バイト	スレーブアドレス(7 ビットアドレス、b0 の値は"0")
	unsigned char far *iic_data_addr	2 バイト	送信/受信バッファのアドレス
	unsigned char iic_num_byte	1 バイト	送信/受信データ数(バイト単位)
構造体の機能説明			
構造体 iic_set_str1~str5(表 4.3を参照)に設定された値を取得し、通信時に使用します。			
構造体のメンバ iic_status の機能説明			
構造体のメンバ iic_status は、通信指示コマンド/通信ステータスを示します。	<コマンド> 0x40 (CWRITEMODE) : マスタ送信コマンド 0x41 (CREADMODE) : マスタ受信コマンド		
・通信開始前 構造体 iic_set_str1~str5 (表 4.3を参照)のメンバ iic_str_command の値(コマンド)が格納されます。	<ステータス> 0x80 (CINWRITE) : 送信動作中 0x81 (CINREAD) : 受信動作中 0x00 (CSUCCESSW) : 送信正常終了 0x01 (CSUCCESSR) : 受信正常終了 0x11 (CBUSBUSY) : IIC バス・エラー		
・通信開始後 iic_status の値が更新され、通信ステータスを示します。	0x12 (CSLAVEBUSY) : スレーブ・ビジー (アドレスに NACK) 0x13 (CNOACK) : データ送信エラー (データに NACK) 0x1f (CCOMERROR) : コマンド・エラー		

表 4.2 定義ファイル名 : r01an3378_src.c (2/4)

変数名	サイズ	使用内容
static unsigned char iic_tx1[TX1_BUFSIZE]	3 バイト	送信バッファ 1
static unsigned char iic_tx2[TX2_BUFSIZE]	2 バイト	送信バッファ 2
static unsigned char iic_tx3[TX3_BUFSIZE]	2 バイト	送信バッファ 3
static unsigned char iic_tx4[TX4_BUFSIZE]	1 バイト	送信バッファ 4
static unsigned char iic_rx1[RX1_BUFSIZE]	8 バイト	受信バッファ 1

表 4.3 定義ファイル名 : r01an3378_src.c (3/4)

変数名	サイズ	使用内容
static iic_set_fcb iic_set_str1, iic_set_str2, iic_set_str3, iic_set_str4, iic_set_str5	全 5 バイト	通信設定パラメータ格納用構造体
構造体の メンバ		
unsigned char iic_str_command	1 バイト	送信/受信コマンド 0x40 (CWRITEMODE) : マスタ送信コマンド 0x41 (CREADMODE) : マスタ受信コマンド
unsigned char iic_str_slave_addr	1 バイト	スレーブアドレス(7ビットアドレス、b0の値は"0")
unsigned char far *iic_str_data_addr	2 バイト	送信/受信バッファのアドレス (送信/受信バッファは、表 4.2を参照)
unsigned char iic_str_num_byte	1 バイト	送信/受信データ数(バイト単位)

表 4.4 定義ファイル名 : r01an3378_src.c (4/4)

変数名	サイズ	使用内容
static unsigned int iic_red_data	2 バイト	赤色の測定データ
static unsigned int iic_green_data	2 バイト	緑色の測定データ
static unsigned int iic_blue_data	2 バイト	青色の測定データ
static unsigned int iic_ir_data	2 バイト	赤外光の測定データ

表 4.5 定義ファイル名 : iic.c

変数名	サイズ	使用内容
static unsigned char iic_status_buf	1 バイト	ステータスバッファ
static unsigned char far *iic_data_pointer	2 バイト	送信/受信バッファへのポインタ
static unsigned char iic_num_byte_buf	1 バイト	送信/受信データ数バッファ
static unsigned char iic_start_cond_flag	1 バイト	スタートコンディションフラグ

表 4.6 定義ファイル名 : tra.c

変数名	サイズ	使用内容
static unsigned int tra_wait_dwncnt	2 バイト	タイマカウントを行う変数

4.2 関数表

宣言	void main (void)		
概要	メイン処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	システムクロック、I ² C バスインタフェースおよびタイマ RA の初期設定後、マスタ送信とマスタ受信を行います。マスタ送受信の完了後は、一定の待ち時間を入れて通信を再開し、マスタ送受信繰り返します。		

宣言	void mcu_init (void)		
概要	システムクロック設定処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	メイン処理から呼び出される関数です。システムクロック(高速オンチップオシレータ)の設定を行います。		

宣言	void port_init (void)		
概要	I/O ポートの初期設定処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	メイン処理から呼び出される関数です。I ² C バスインタフェースを使用するための I/O ポートの初期設定を行います。		

宣言	void iic_init (void)		
概要	I ² C 初期設定処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	メイン処理から呼び出される関数です。I ² C バスインタフェースを使用するための SFR 初期設定を行います。		

宣言	void tra_init (void)		
概要	タイマ RA 初期設定処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	メイン処理から呼び出される関数です。タイマ RA を使用するための SFR 初期設定を行います。		

宣言	unsigned char iic_sensor_init (unsigned char far *iic_set_str)		
概要	カラーセンサの初期化処理		
引数	引数名	意味	
	unsigned char far *iic_set_str	通信設定パラメータ格納用構造体へのポインタ	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
戻り値	型	値	意味
	unsigned char	0	通信エラー
		1	通信成功
機能説明	<p>メイン処理から呼び出される関数です。カラーセンサを初期化するための処理を行います。</p> <p>iic_copy_parameter 関数を呼び出して送信データを取得し、iic_start 関数を呼び出して通信を開始します。通信中は、構造体 iic_comstat の通信ステータスを確認し、データ送信の完了まで待ちます。通信を終了するときは、iic_stop_cond 関数を呼び出して通信終了処理を行います。</p> <p>通信ステータスの確認で、通信エラーが発生した場合、iic_stop_cond 関数を呼び出して通信を停止します。</p>		

宣言	static void iic_copy_parameter (unsigned char far *iic_set_str)		
概要	通信パラメータのコピー処理		
引数	引数名	意味	
	unsigned char far *iic_set_str	通信設定パラメータ格納用構造体へのポインタ	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>通信開始前に、呼び出される関数です。</p> <p>通信に必要なパラメータを取得するために、通信用データを(構造体)iic_comstat へコピーします。</p>		

宣言	static void iic_start (void)		
概要	I ² C 通信開始処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
	static unsigned char iic_status_buf	通信時の通信ステータスを格納するバッファ	
	static unsigned char far *iic_data_pointer	通信時に送受信データを示すポインタ	
	static unsigned char iic_num_byte_buf	通信時に残りのデータ数を示すバッファ	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>通信開始時に、呼び出される関数です。通信を開始するための処理を行います。iic_start_cond 関数を呼び出してスタートコンディションを発行した後、スレーブアドレスを送信します。</p> <p>この関数の先頭では、外部デバイスにバスが占有されていないこと、送信/受信コマンドおよび変数 iic_num_byte_buf を確認します。この確認の結果、エラーが発生している場合は、通信開始処理を行いません。</p>		

宣言	static void iic_start_cond (void)		
概要	スタートコンディション発行処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	static unsigned char iic_start_cond_flag	スタートコンディションフラグ	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>I²C 通信開始処理から呼び出される関数です。</p> <p>スタートコンディションを発行し、スタートコンディションフラグを設定します。</p>		

宣言	static void iic_stop_cond (void)		
概要	ストップコンディション発行処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>通信を停止するときに、呼び出される関数です。ストップコンディションを発行するための処理を行います。</p> <p>この関数の先頭では、バス状態を確認します。</p> <ul style="list-style-type: none"> ・バスビジーの場合 ストップコンディションの発行処理を行います。 ・バスフリーの場合 ストップコンディションの発行処理を行いません。 		

宣言	unsigned char iic_sensor_reset (unsigned char far *iic_set_str, unsigned char far *iic_set_str_2nd)		
概要	カラーセンサのリセット処理		
引数	引数名	意味	
	unsigned char far *iic_set_str	通信設定パラメータ格納用構造体へのポインタ	
	unsigned char far *iic_set_str_2nd	通信設定パラメータ格納用構造体へのポインタ	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
戻り値	型	値	意味
	unsigned char	0	通信エラー
		1	通信成功
機能説明	<p>メイン処理から呼び出される関数です。カラーセンサをリセットするための処理を行います。</p> <p>iic_copy_parameter 関数を呼び出して送信データを取得し、iic_start 関数を呼び出して通信を開始します。通信中は、構造体 iic_comstat の通信ステータスを確認し、データ送信の完了まで待ちます。通信を終了するときは、iic_stop_cond 関数を呼び出して通信終了処理を行います。</p> <p>通信ステータスの確認で、通信エラーが発生した場合は、iic_stop_cond 関数を呼び出して通信を停止します。</p>		

宣言	void tra_measure_wait (void)		
概要	光強度の計測完了待ち処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	static unsigned int tra_wait_dwncnt	タイマカウンタ	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>メイン処理から呼び出される関数です。カラーセンサが光強度を計測している間、待機するための処理を行います。</p> <p>変数 tra_wait_dwncnt にタイマのカウンタ値を設定し、tra_start 関数を呼び出してタイマ RA の動作を開始します。タイマ RA の動作中は、タイマ RA カウンタステータスフラグ(TCSTF)の確認を行い、タイマ RA が動作停止するまで待ちます。</p> <p>変数 tra_wait_dwncnt の更新およびタイマ RA の停止は、タイマ RA の割り込み処理により行われます。</p>		

宣言	static void tra_start (void)		
概要	タイマ RA 動作開始処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>タイマ RA を動作させるときに呼び出される関数です。</p> <p>タイマ RA を動作させるための処理を行います。</p>		

宣言	unsigned char iic_sensor_read (unsigned char far *iic_set_str, unsigned char far *iic_set_str_2nd)		
概要	測定データの読み出し処理		
引数	引数名	意味	
	unsigned char far *iic_set_str	通信設定パラメータ格納用構造体へのポインタ	
	unsigned char far *iic_set_str_2nd	通信設定パラメータ格納用構造体へのポインタ	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
戻り値	型	値	意味
	unsigned char	0 1	通信エラー 通信成功
機能説明	<p>メイン処理から呼び出される関数です。カラーセンサの測定データを読み出すための処理を行います。</p> <p>iic_copy_parameter 関数を呼び出して送信データを取得し、iic_start 関数を呼び出して通信を開始します。通信中は、構造体 iic_comstat の通信ステータスを確認し、データ送信/受信の完了まで待ちます。最終バイトの受信後に I²C バスインタフェース割り込み処理によりストップコンディションを発行し、通信を終了しています。</p> <p>通信ステータスの確認で、通信エラーが発生した場合、iic_stop_cond 関数を呼び出して通信を停止します。</p>		

宣言	void iic_measure_result (unsigned char far *iic_rx, unsigned char iic_rx_size, unsigned int far *iic_red_data, unsigned int far *iic_green_data, unsigned int far *iic_blue_data, unsigned int far *iic_ir_data)		
概要	測定結果の処理		
引数	引数名	意味	
	unsigned char far *iic_rx	受信バッファへのポインタ	
	unsigned char iic_rx_size	受信バッファのサイズ	
	unsigned int far *iic_red_data	赤色光の測定データを格納する変数へのポインタ	
	unsigned int far *iic_green_data	緑色光の測定データを格納する変数へのポインタ	
	unsigned int far *iic_blue_data	青色光の測定データを格納する変数へのポインタ	
	unsigned int far * iic_ir_data	赤外光の測定データを格納する変数へのポインタ	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>メイン処理から呼び出される関数です。受信バッファに格納された受信データを各色の測定データを示す変数へ格納するための処理を行います。</p>		

宣言	void tra_next_com_wait (void)		
概要	通信再開待ち処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	static unsigned int tra_wait_dwncnt	タイマカウンタ	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>メイン処理から呼び出される関数です。次の通信を再開させるまでの間、待機するための処理を行います。</p> <p>変数 tra_wait_dwncnt にタイマのカウント値を設定し、tra_start 関数を呼び出してタイマ RA の動作を開始します。タイマ RA の動作中は、タイマ RA カウントステータスフラグ(TCSTF)の確認を行い、タイマ RA が動作停止するまで待ちます。</p> <p>変数 tra_wait_dwncnt の更新およびタイマ RA の停止は、タイマ RA の割り込み処理により行われます。</p>		

宣言	void iic_error (void)		
概要	I ² C 通信エラー処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	なし	-	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>通信エラーが発生した場合に、メイン処理から呼び出される関数です。処理を何もせずに終了します。エラーに対する処理を行う場合は、プログラムを追加してください。</p>		

宣言	void iic_int (void)		
概要	I ² C バスインタフェース割り込み処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
	static unsigned char iic_status_buf	通信時の通信ステータスを格納するバッファ	
	static unsigned char iic_start_cond_flag	スタートコンディションフラグ	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>SCL クロックの 9 ビット目の立ち上がりで割り込みが発生します。</p> <p>この関数の先頭では、変数 iic_status_buf でデータ送信の直前にスタートコンディションが発行されているかを確認します。</p> <p>スタートコンディションが発行されている場合、送信モードの時に iic_data_trs_int 関数を、受信モードの時に iic_set_rcv_int 関数を呼び出します。</p> <p>スタートコンディションが発行されていない場合、送信モードの時に iic_cont_data_trs_int 関数を、受信モードの時に iic_data_rcv_int 関数を呼び出します。</p> <p>通信エラーが発生した場合は、iic_error_exit_int 関数を呼び出して通信を終了します。</p>		

宣言	static void iic_data_trs_int (void)		
概要	データ送信処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	static unsigned char far *iic_data_pointer	通信時に送受信データを示すポインタ	
戻り値	型	値	意味
	なし	-	-
機能説明	I ² C バスインタフェース割り込み処理から呼び出される関数です。 ポインタ iic_data_pointer が指す送信バッファ内のデータを送信します。 また、次の送信を行うためにポインタ iic_data_pointer の更新を行います。		

宣言	static void iic_cont_data_trs_int (void)		
概要	連続データ送信処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
	static unsigned char iic_status_buf	通信時の通信ステータスを格納するバッファ	
	static unsigned char iic_num_byte_buf	通信時に残りのデータ数を示すバッファ	
戻り値	型	値	意味
	なし	-	-
機能説明	I ² C バスインタフェース割り込み処理から呼び出される関数です。 この関数の先頭では、ACK が検出されているかを確認します。 ・ ACK 検出の場合 変数 iic_num_byte_buf の更新後に、残りの送信データ数を確認します。残りの送信データがある時、iic_data_trs_int 関数を呼び出してデータ送信処理を行います。残りの送信データがない時、通信の終了を示すために構造体 iic_comstat の通信ステータスを更新します。 ・ NACK 検出の場合 iic_error_exit_int 関数を呼び出して通信を終了します。		

宣言	static void iic_set_rcv_int (void)		
概要	受信設定処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	static unsigned char iic_num_byte_buf	通信時に残りのデータ数を示すバッファ	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>I²C バスインタフェース割り込み処理から呼び出される関数です。マスタ送信モードからマスタ受信モードへ切り替えるための設定を行い、受信データフル割り込み要求を許可します。</p> <p>この関数内では、変数 iic_num_byte_buf で受信予定の受信データ数を確認しています。</p> <ul style="list-style-type: none"> ・受信データ数が 1 バイトの場合 アクリッジのタイミングで "1" (NACK) を送出する設定および次のデータ受信後の受信動作を禁止する設定を行います。 ・受信バイト数が 2 バイト以上の場合 アクリッジのタイミングで "0" (ACK) を送出する設定および次のデータ受信後の受信動作を継続する設定を行います。 		

宣言	static void iic_data_rcv_int (void)		
概要	データ受信処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
	static unsigned char far *iic_data_pointer	通信時に送受信データを示すポインタ	
	static unsigned char iic_num_byte_buf	通信時に残りのデータ数を示すバッファ	
戻り値	型	値	意味
	なし	-	-
機能説明	<p>I²C バスインタフェース割り込み処理から呼び出される関数です。</p> <p>ポインタ iic_data_pointer が指す受信バッファ内の領域へ受信データを格納し、次の受信を行うためにポインタ iic_data_pointer の更新を行います。</p> <p>また、変数 iic_num_byte_buf の更新を行い、残りの受信データ数を確認しています。</p> <ul style="list-style-type: none"> ・残りの受信データ数が 0 バイトの場合 通信を終了するためにストップコンディションの発行、受信データフル割り込み要求の禁止および構造体 iic_comstat の通信ステータスの更新を行います。 ・残りの受信データ数が 1 バイトの場合 アクリッジのタイミングで "1" (NACK) を送出する設定および次のデータ受信後の受信動作を禁止する設定を行います。 		

宣言	static void iic_error_exit_int (void)		
概要	通信エラー終了処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	(構造体)iic_comstat	通信ステータス	
	static unsigned char iic_status_buf	通信時の通信ステータスを格納するバッファ	
	static unsigned char iic_num_byte_buf	通信時に残りのデータ数を示すバッファ	
戻り値	型	値	意味
	なし	-	-
機能説明	I ² C バスインタフェース割り込み処理の実行中に通信エラーが発生した場合、呼び出される関数です。通信を終了するために構造体 iic_comstat の通信ステータスおよび送信/受信データ数を更新します。		

宣言	void tra_int (void)		
概要	タイマ RA 割り込み処理		
引数	引数名	意味	
	なし	-	
使用変数 (グローバル)	変数名	使用内容	
	static unsigned int tra_wait_dwncnt	タイマカウンタ	
戻り値	型	値	意味
	なし	-	-
機能説明	タイマ RA のアンダフローにより 25ms の間隔で割り込みが発生します。この関数では変数 tra_wait_dwncnt をデクリメントし、その結果を確認します。変数 tra_wait_dwncnt の値が"0"の場合、タイマ RA のカウンタを停止します。		

4.3 メイン処理

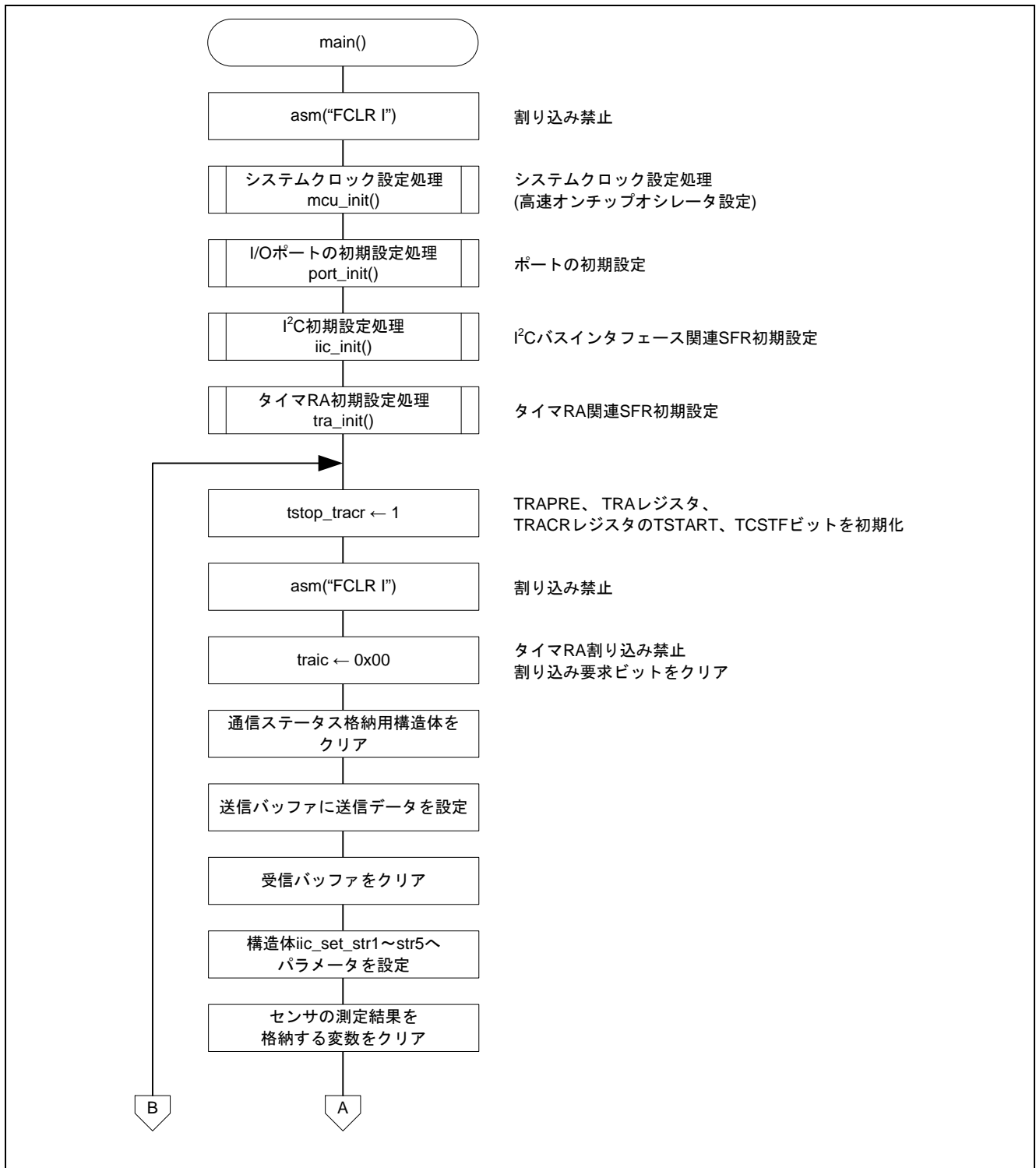


図 4.1 メイン処理 (1/2)

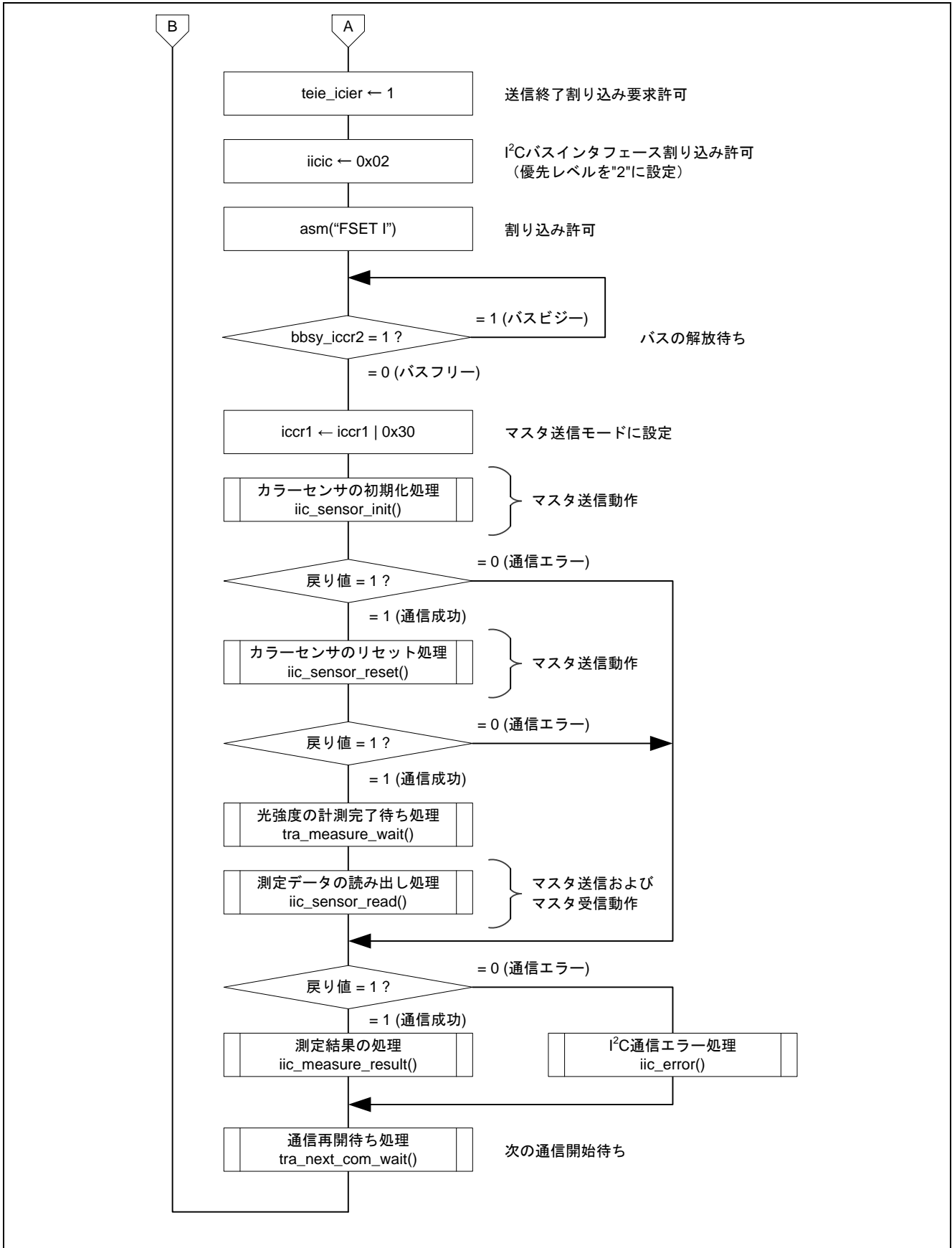


図 4.2 メイン処理 (2/2)

4.4 システムクロック設定処理

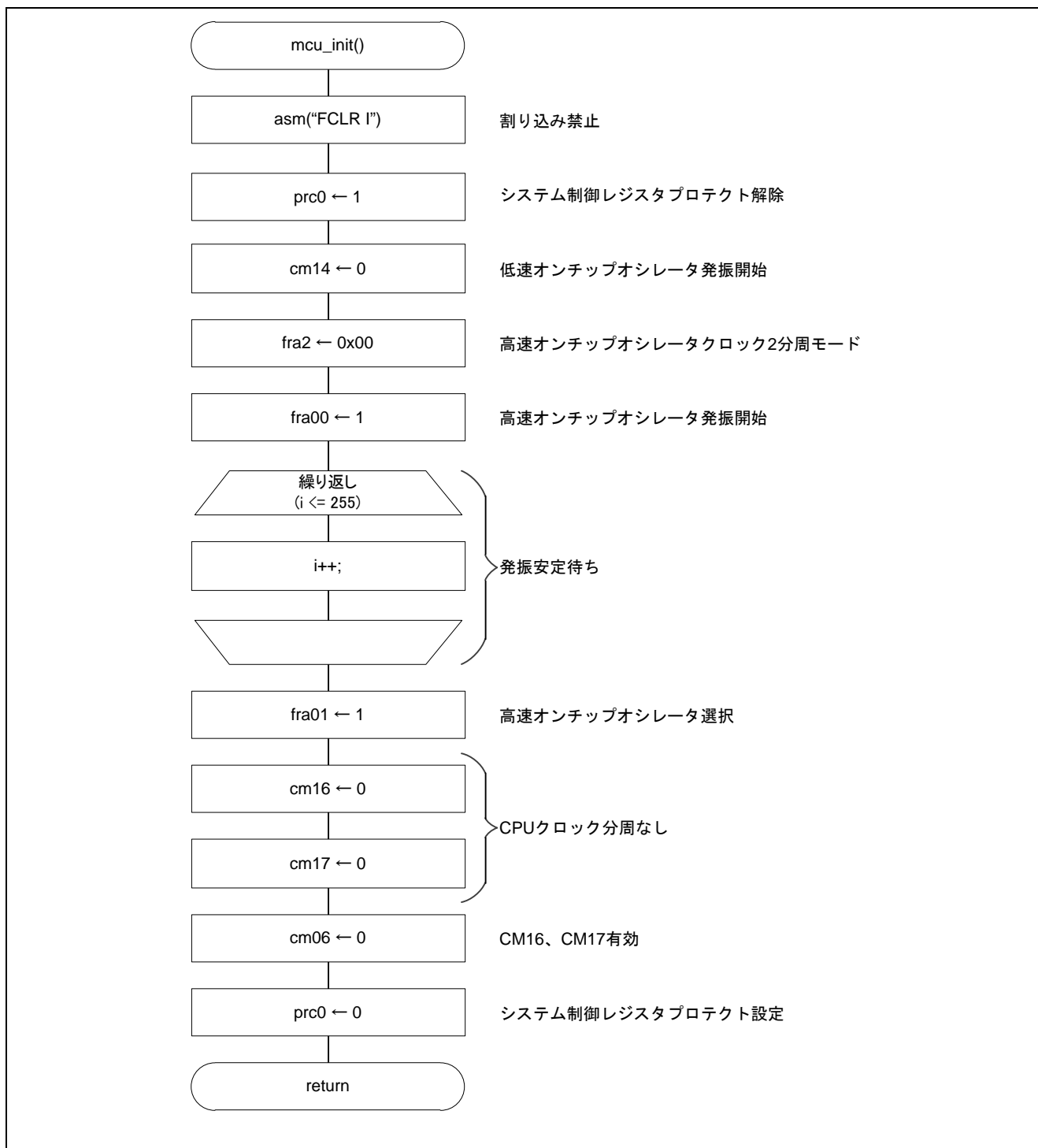


図 4.3 システムクロック設定処理

4.5 I/O ポートの初期設定処理

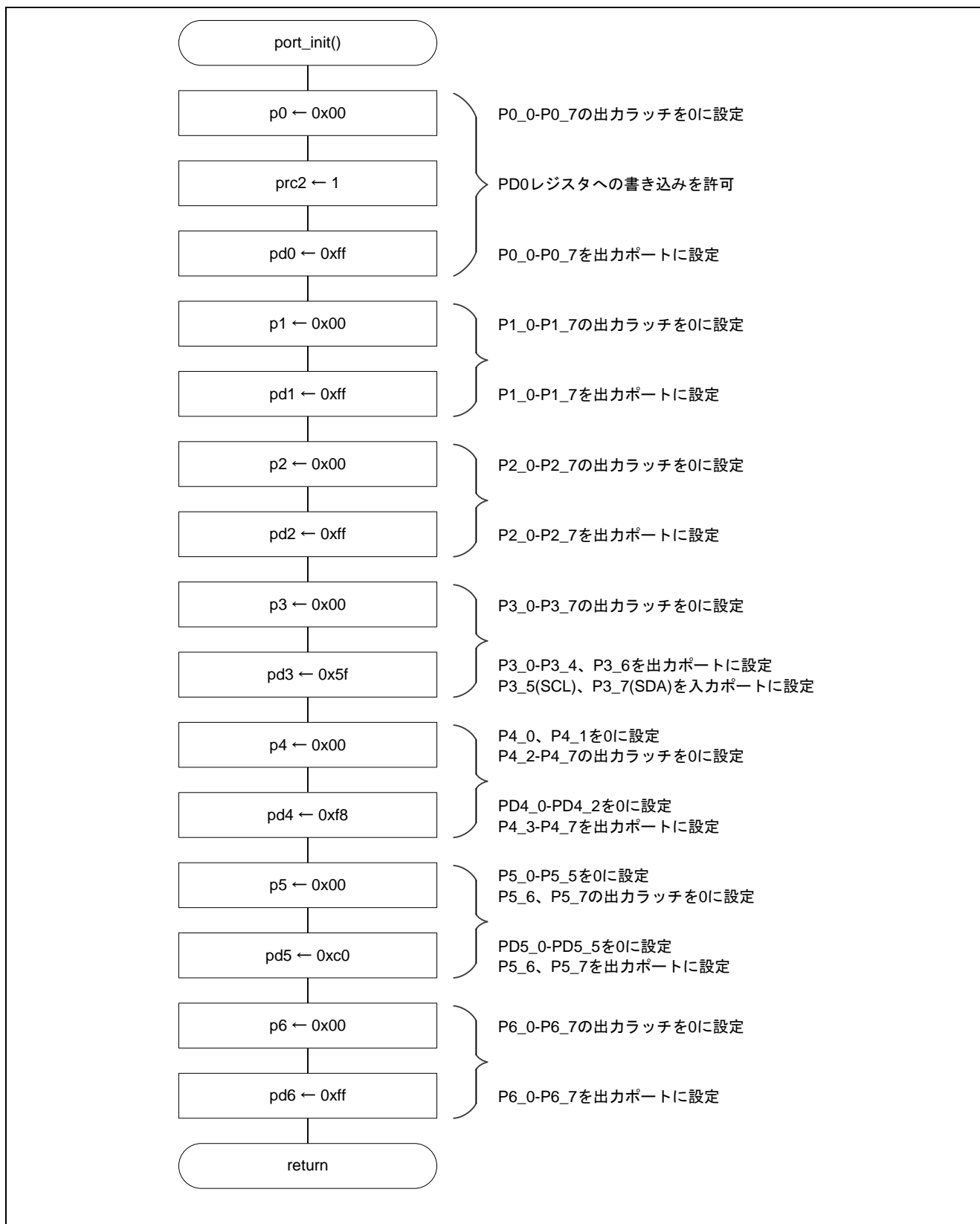
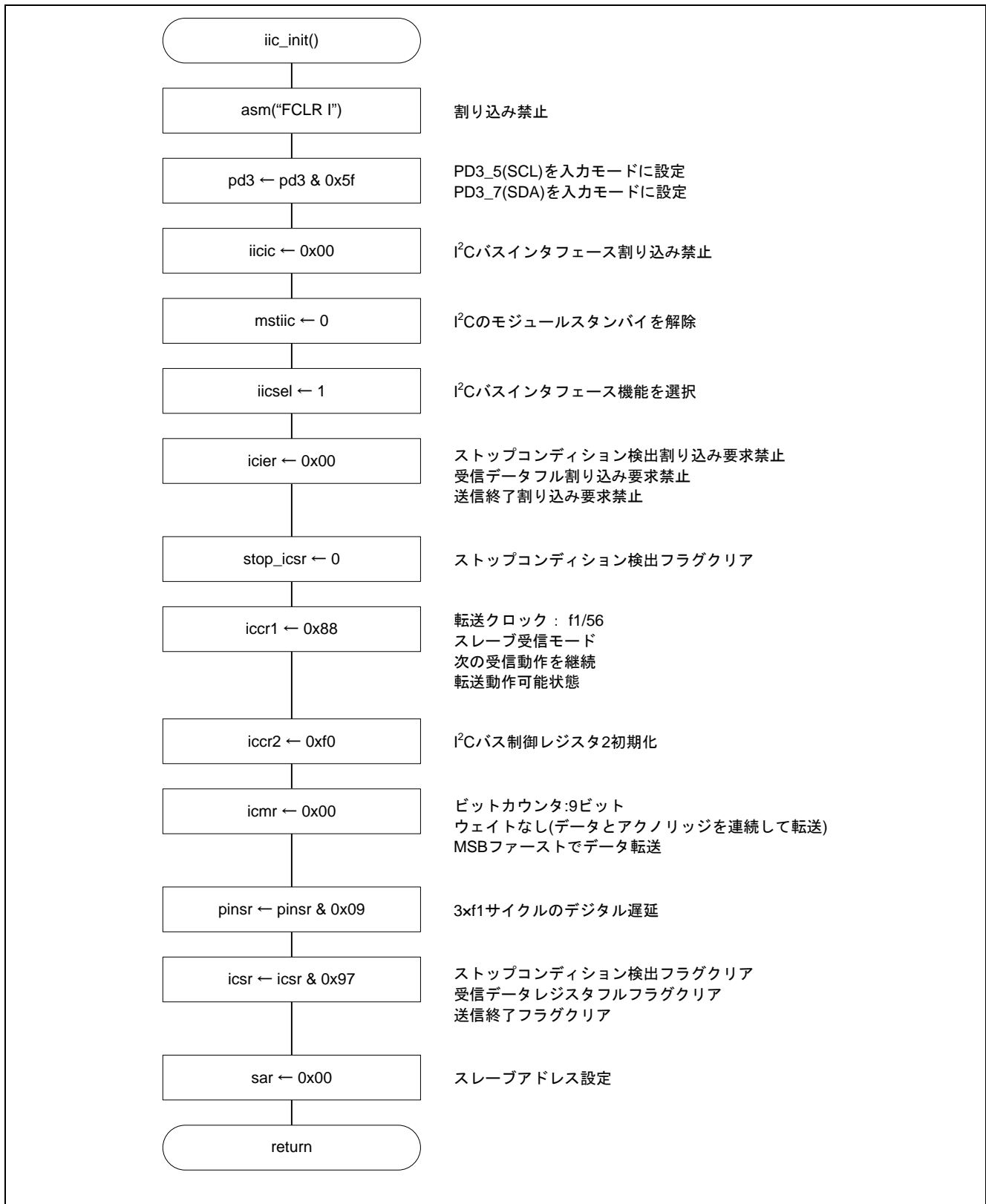


図 4.4 I/O ポートの初期設定処理

4.6 I²C 初期設定処理図 4.5 I²C 初期設定処理

4.7 タイマ RA 初期設定処理

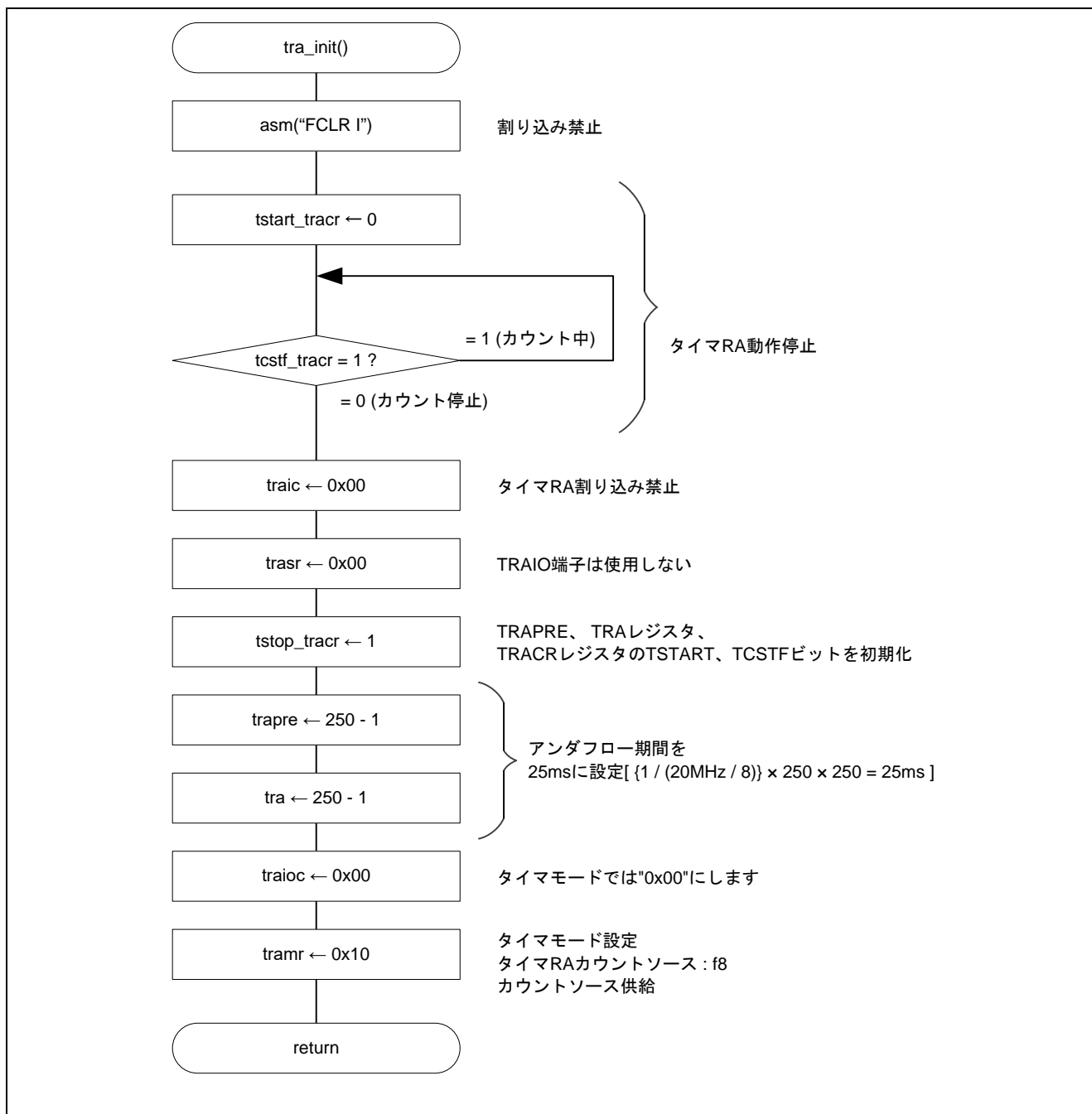


図 4.6 タイマ RA 初期設定処理

4.8 カラーセンサの初期化処理

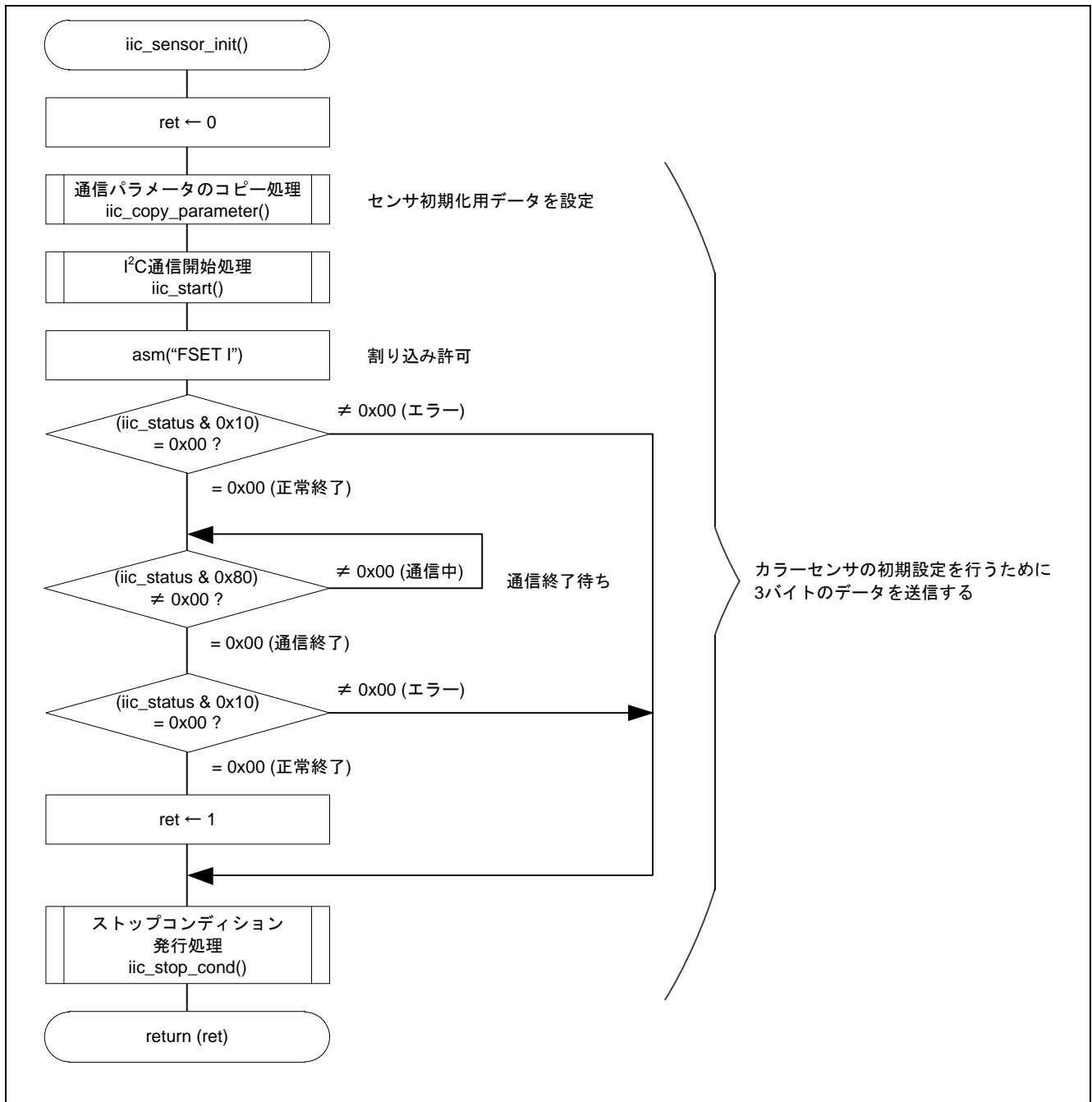


図 4.7 カラーセンサの初期化処理

4.9 通信パラメータのコピー処理

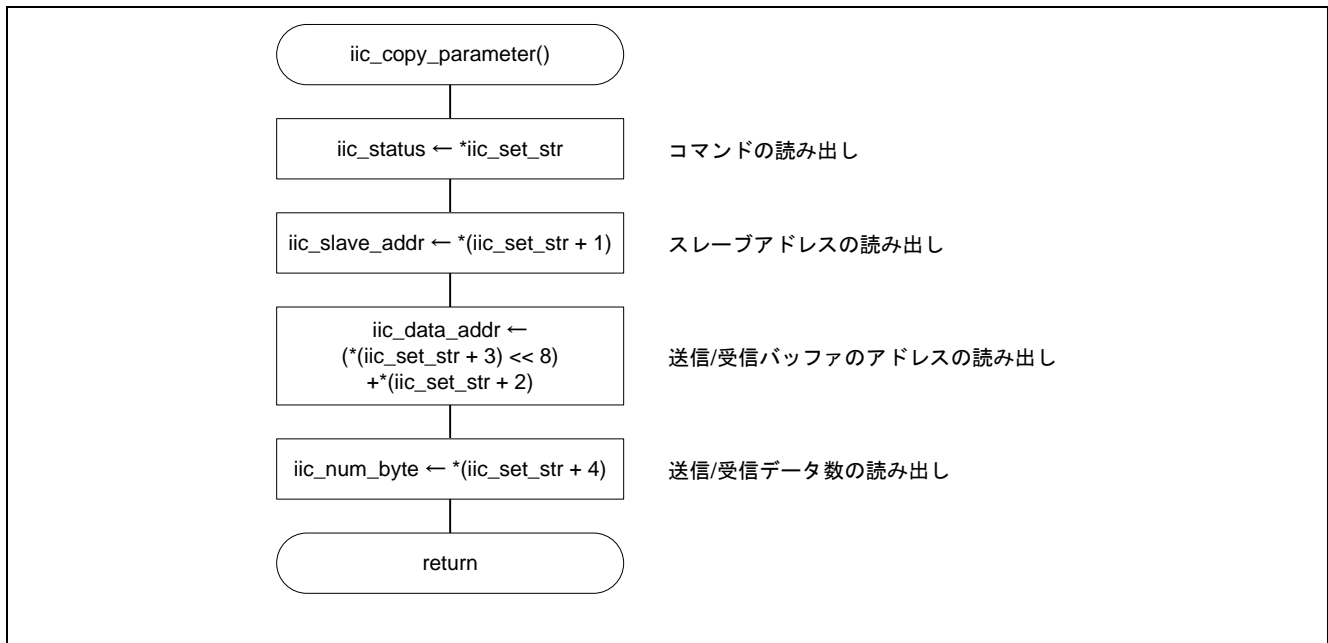
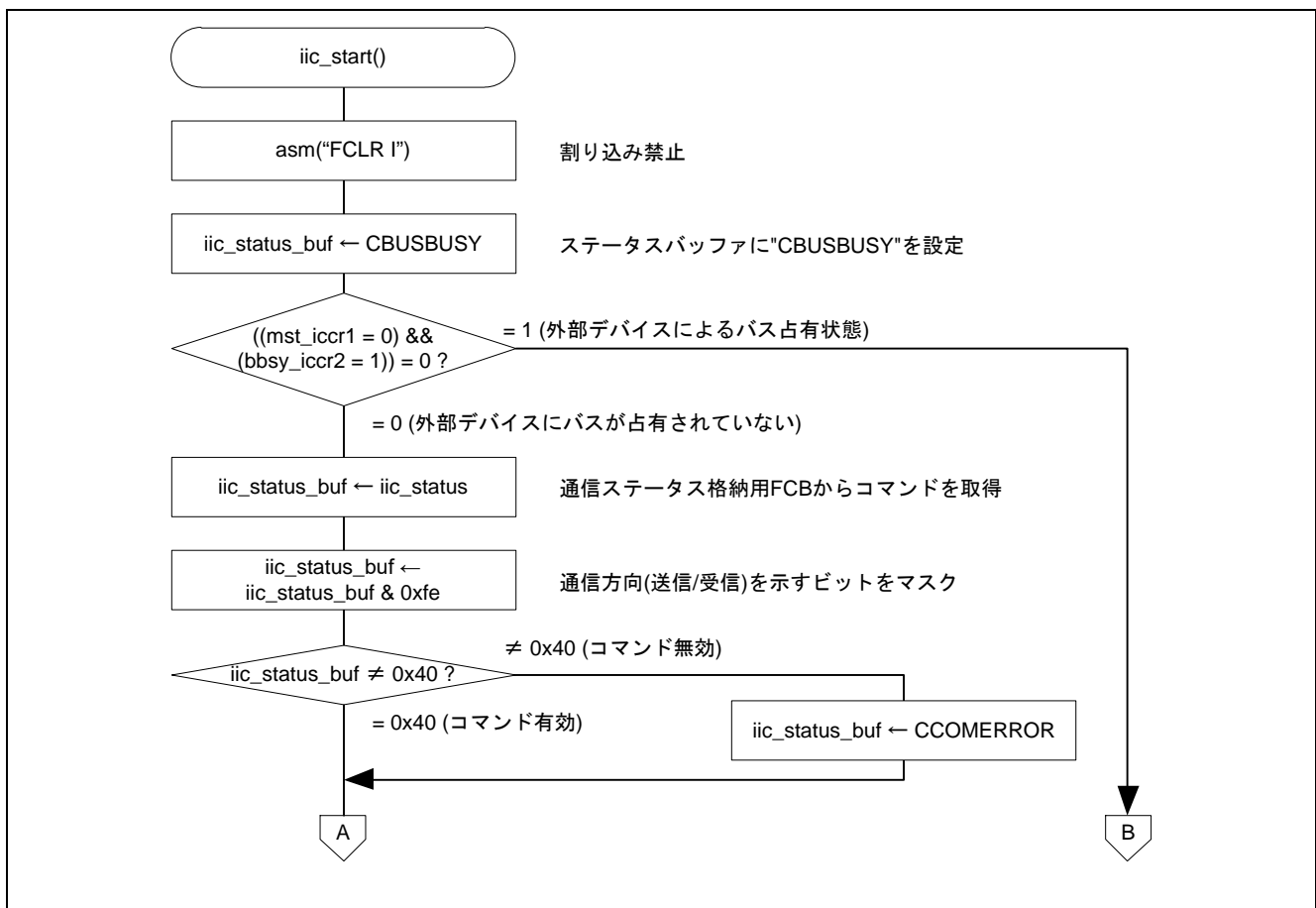


図 4.8 通信パラメータのコピー処理

4.10 I²C 通信開始処理図 4.9 I²C 通信開始処理 (1/2)

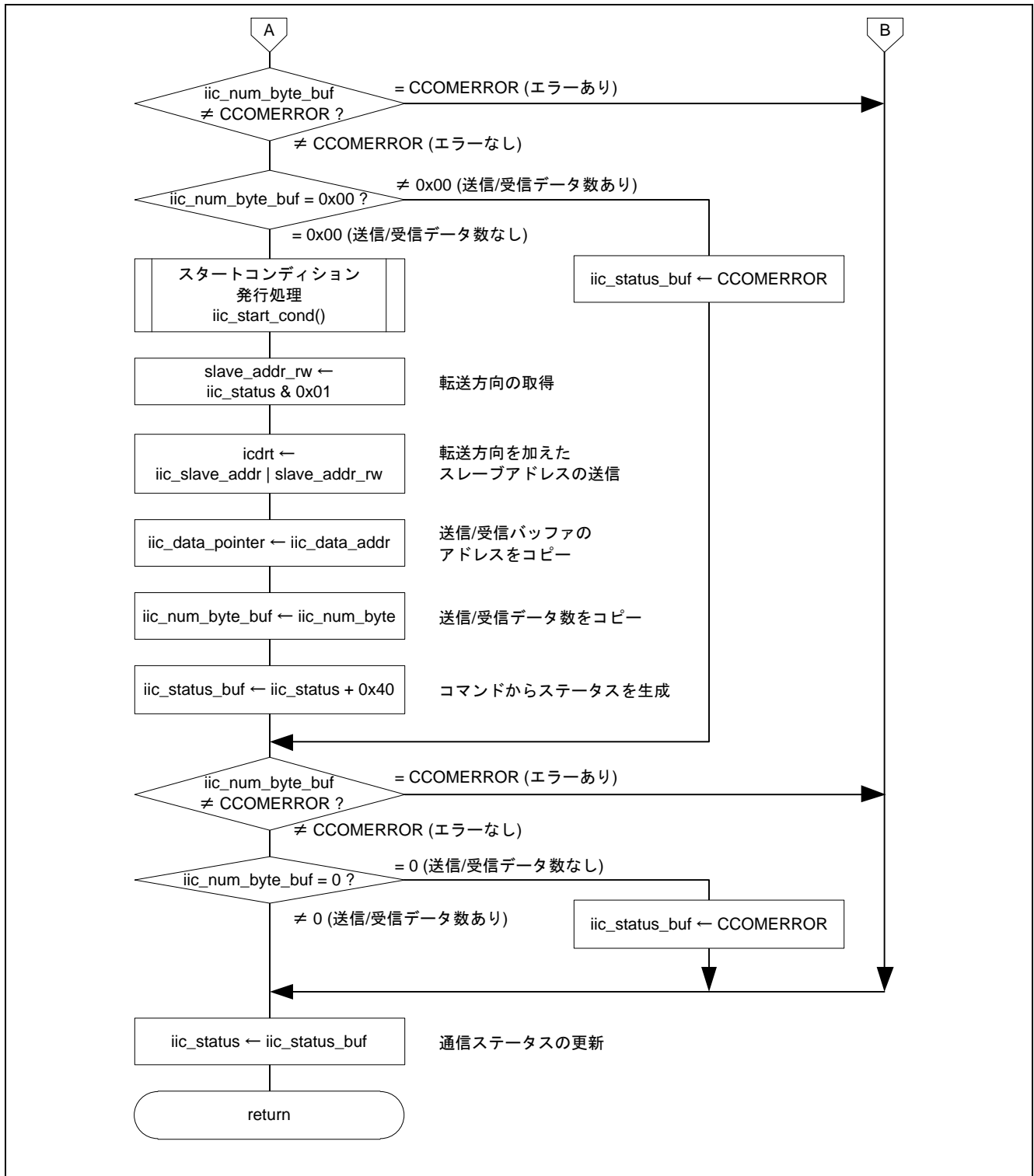


図 4.10 I²C 通信開始処理 (2/2)

4.11 スタートコンディション発行処理

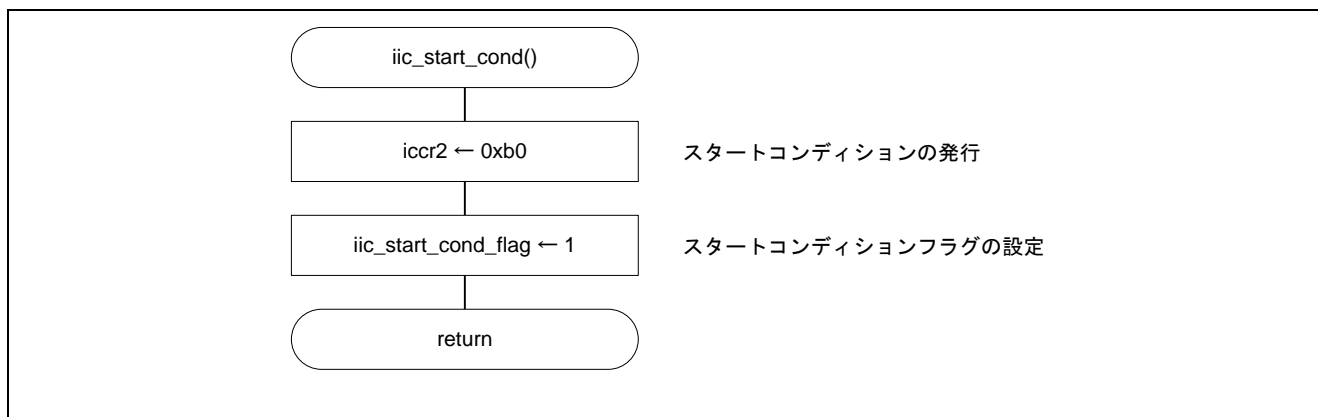


図 4.11 スタートコンディション発行処理

4.12 ストップコンディション発行処理

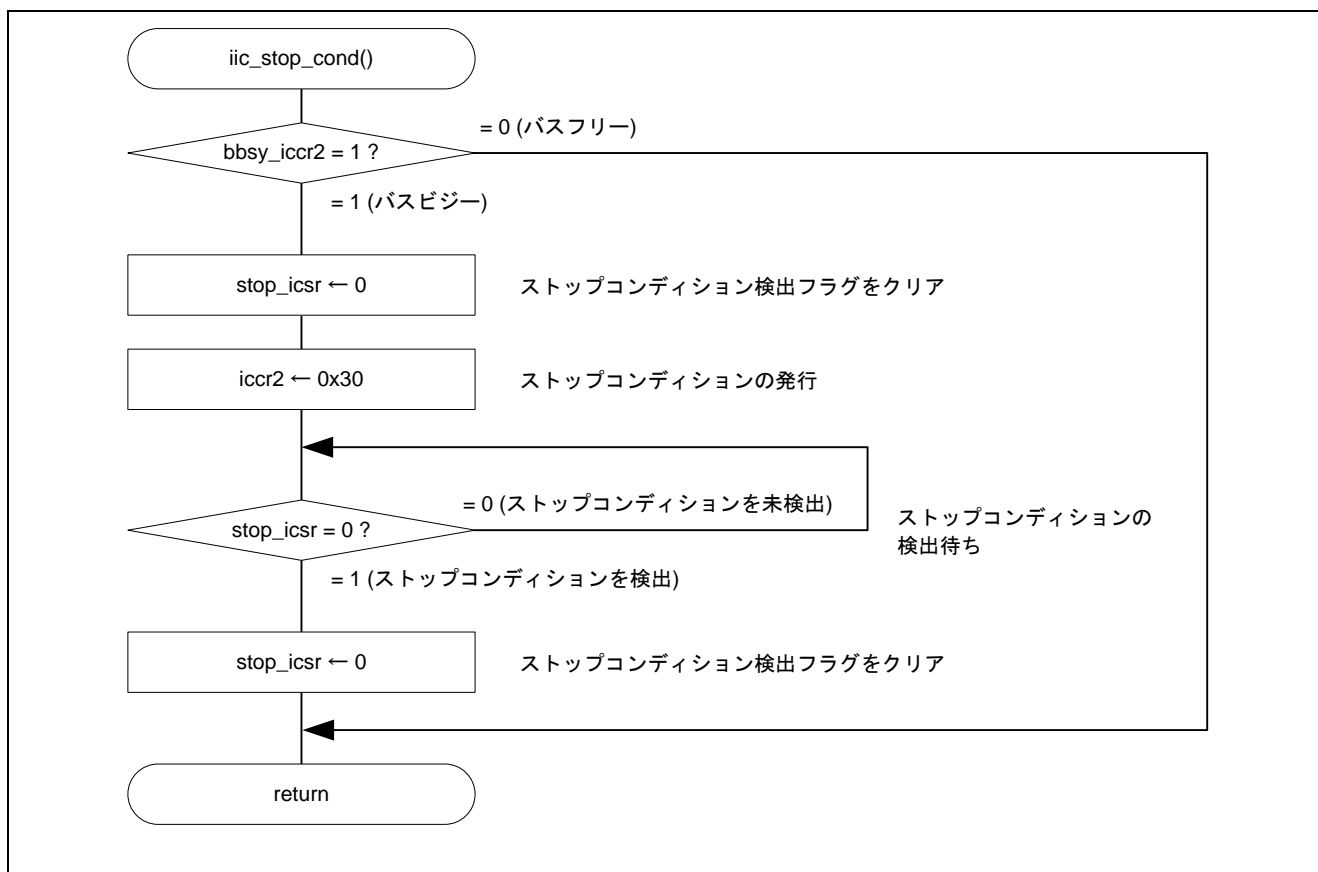


図 4.12 ストップコンディション発行処理

4.13 カラーセンサのリセット処理

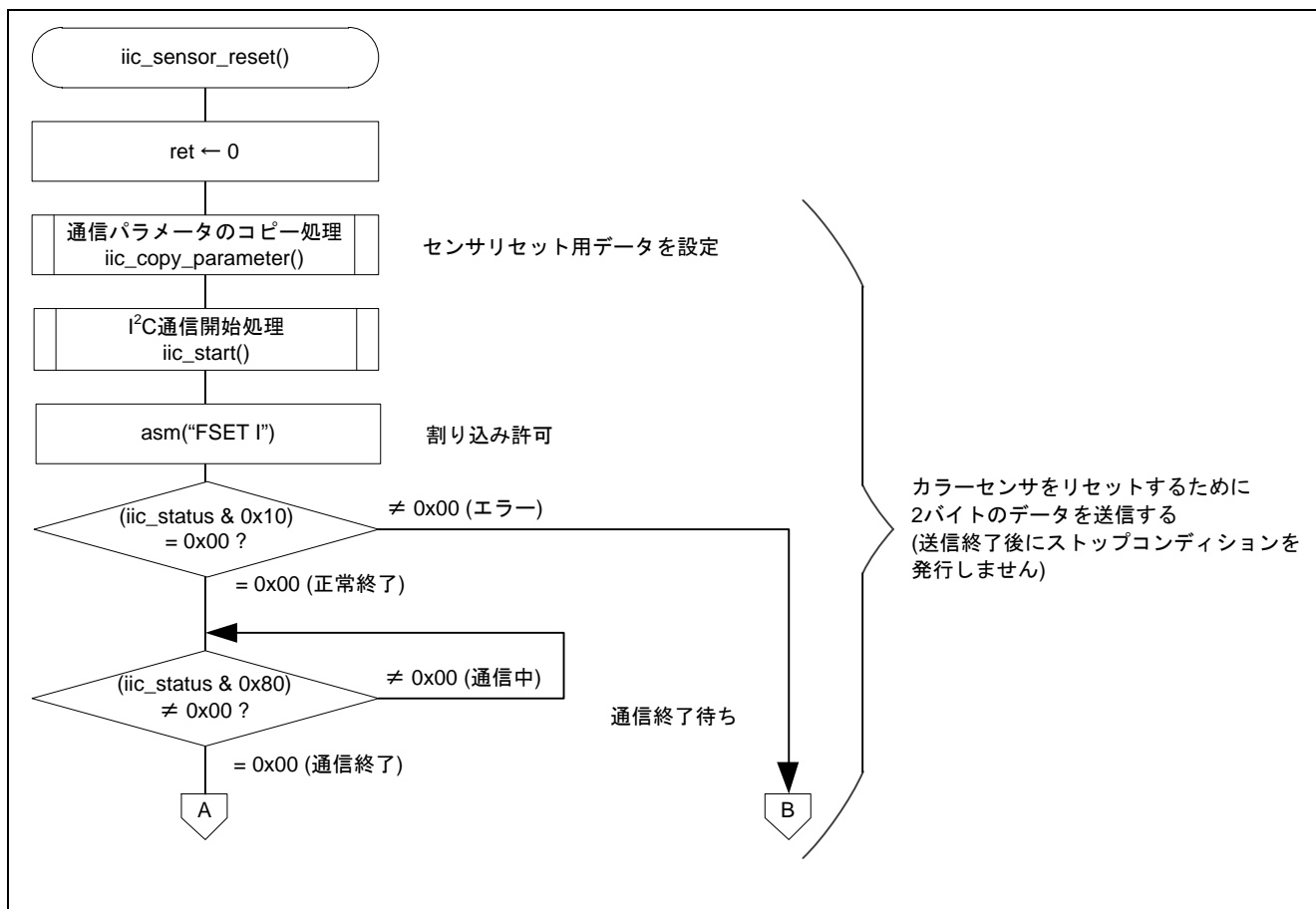


図 4.13 カラーセンサのリセット処理 (1/2)

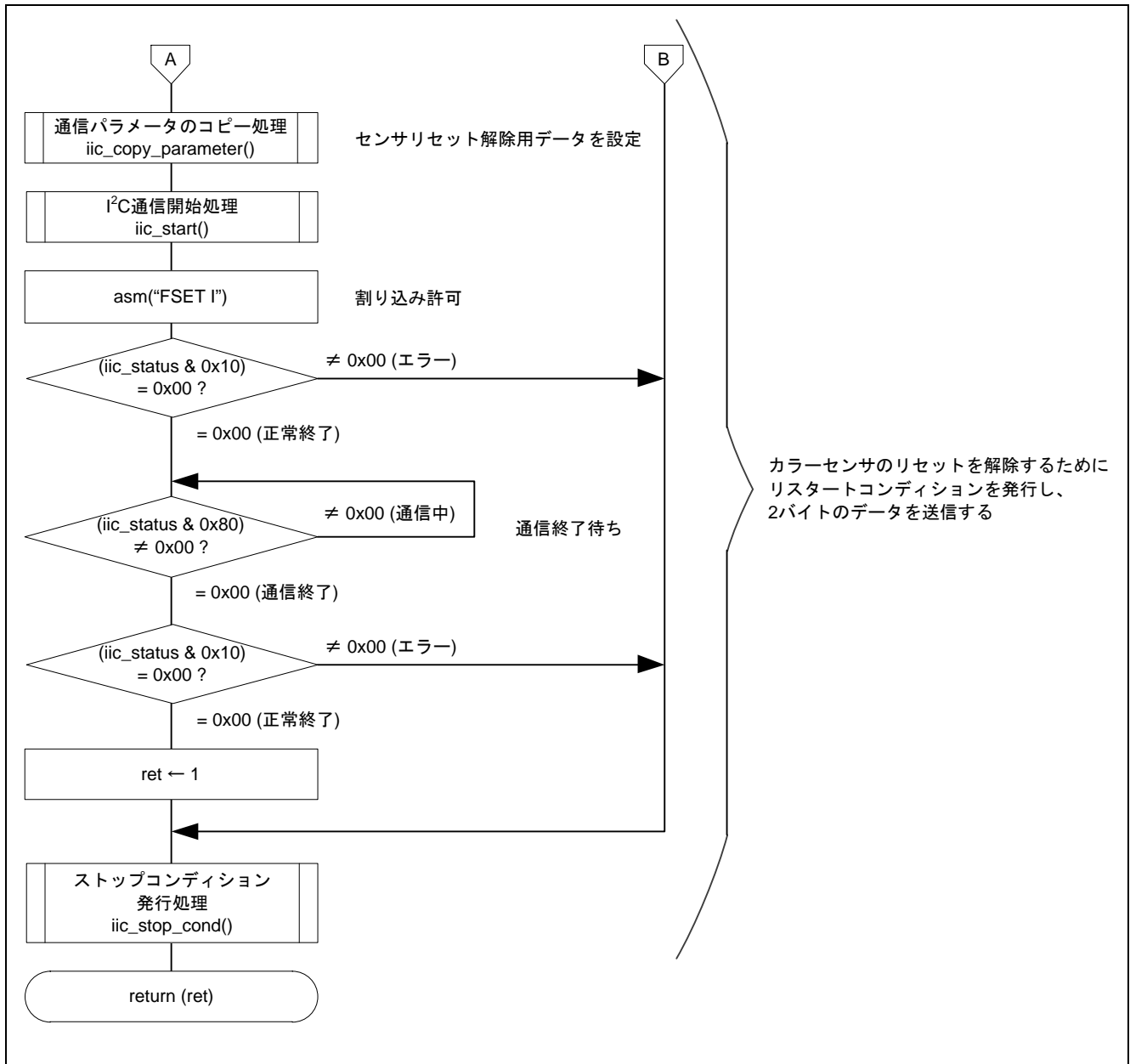


図 4.14 カラーセンサのリセット処理 (2/2)

4.14 光強度の計測完了待ち処理

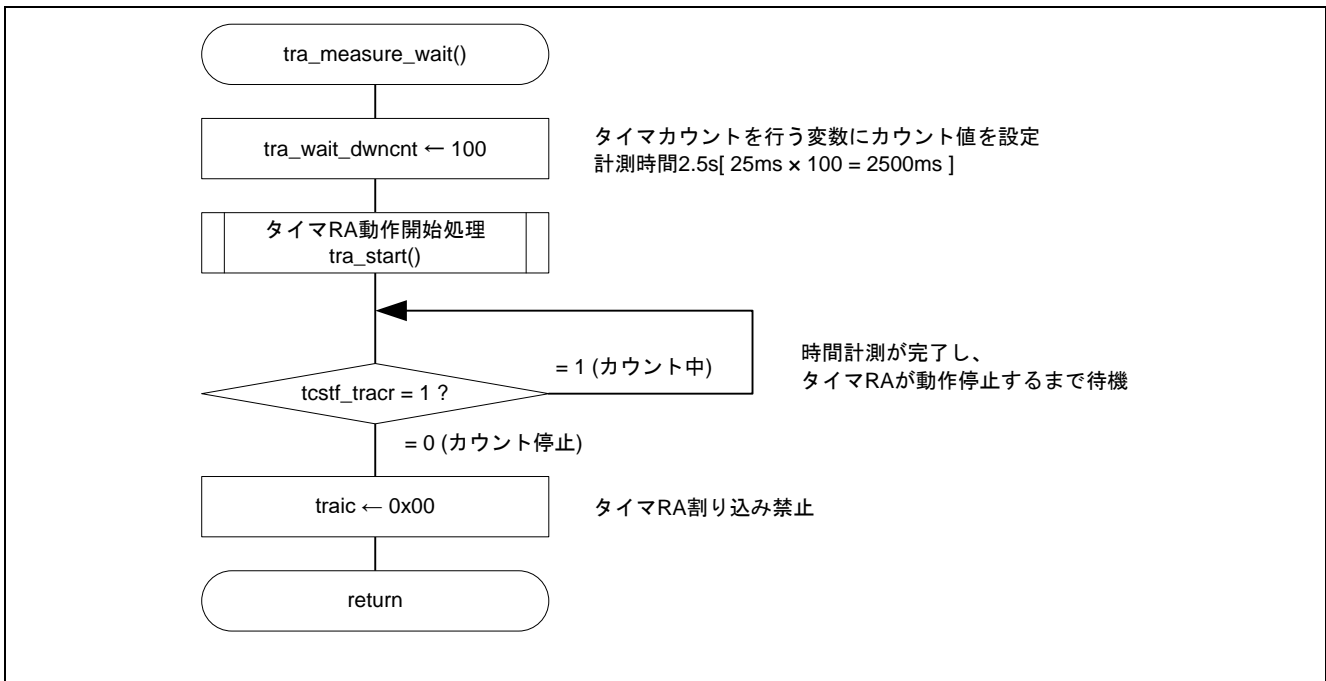


図 4.15 光強度の計測完了待ち処理

4.15 タイマ RA 動作開始処理

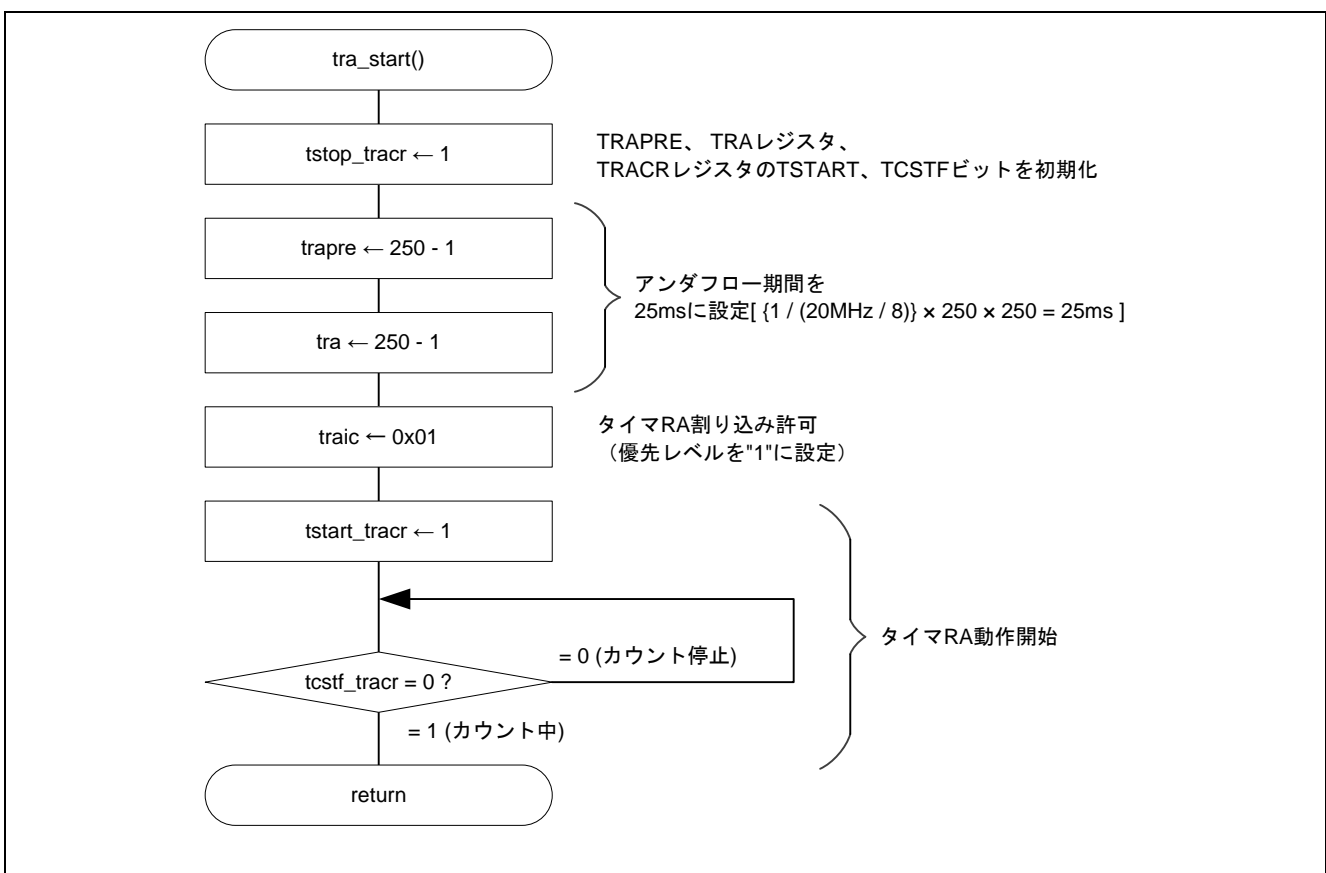


図 4.16 タイマ RA 動作開始処理

4.16 測定データの読み出し処理

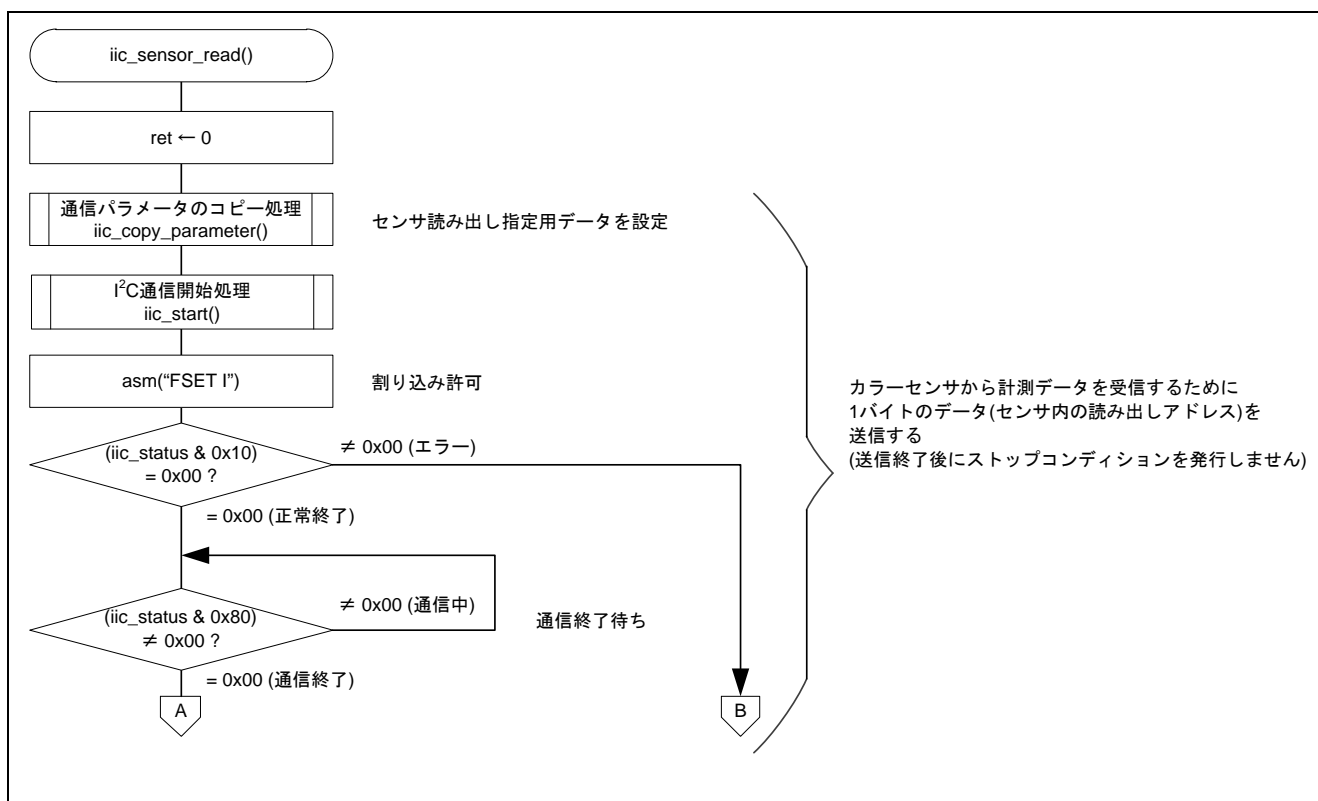


図 4.17 測定データの読み出し処理 (1/2)

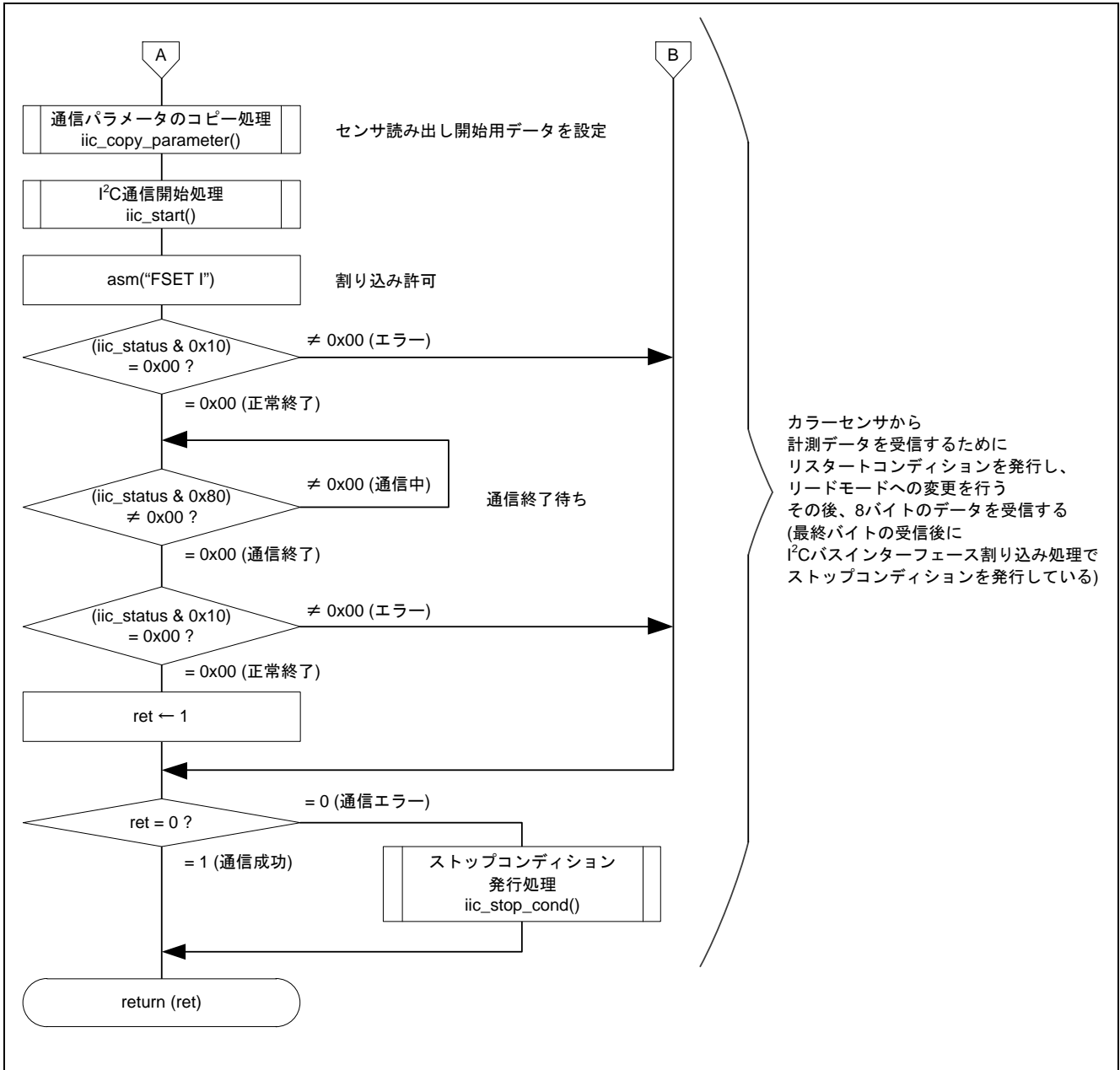


図 4.18 測定データの読み出し処理 (2/2)

4.17 測定結果の処理

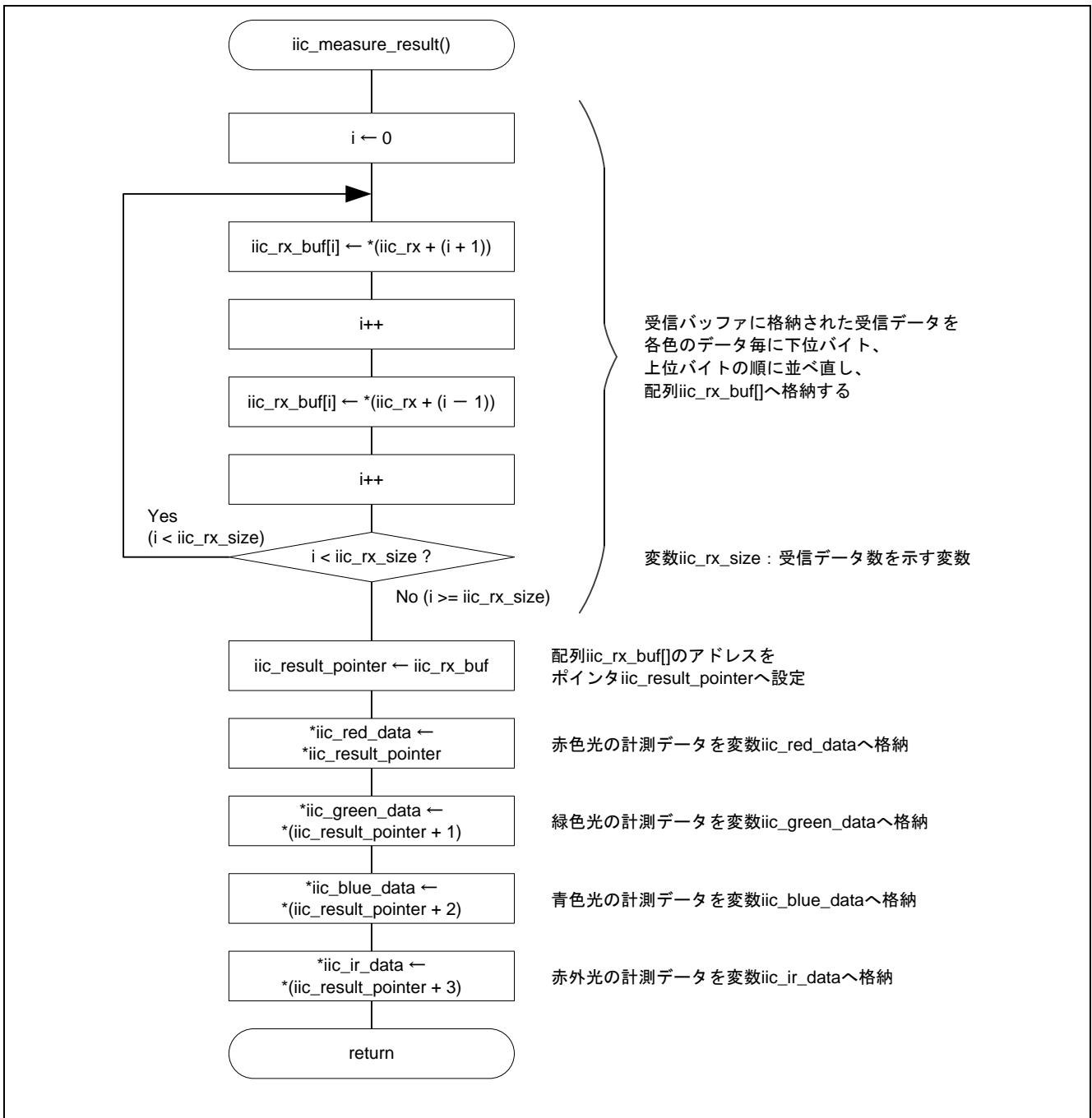


図 4.19 測定結果の処理

4.18 通信再開待ち処理

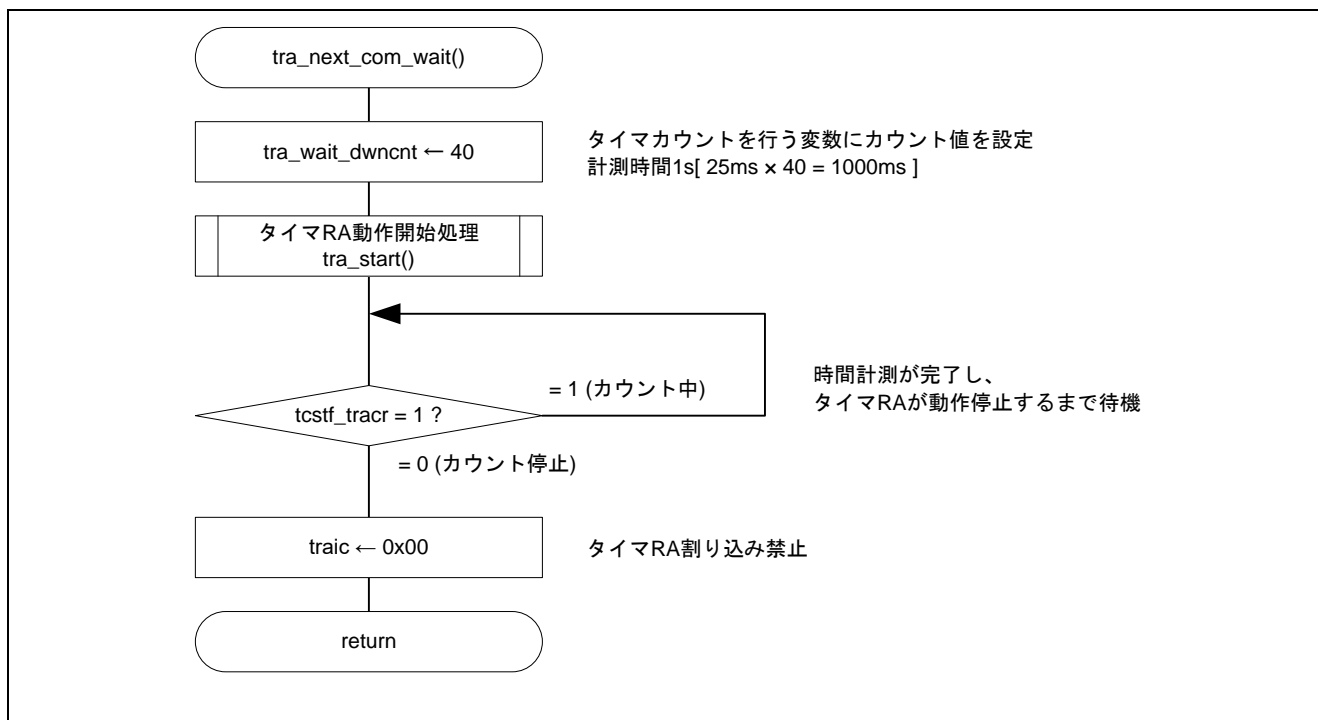
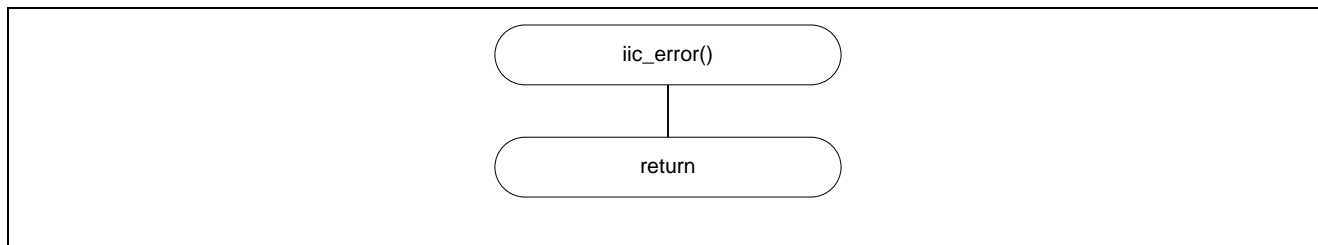


図 4.20 通信再開待ち処理

4.19 I²C 通信エラー処理図 4.21 I²C 通信エラー処理

4.20 I²C バスインタフェース割り込み処理

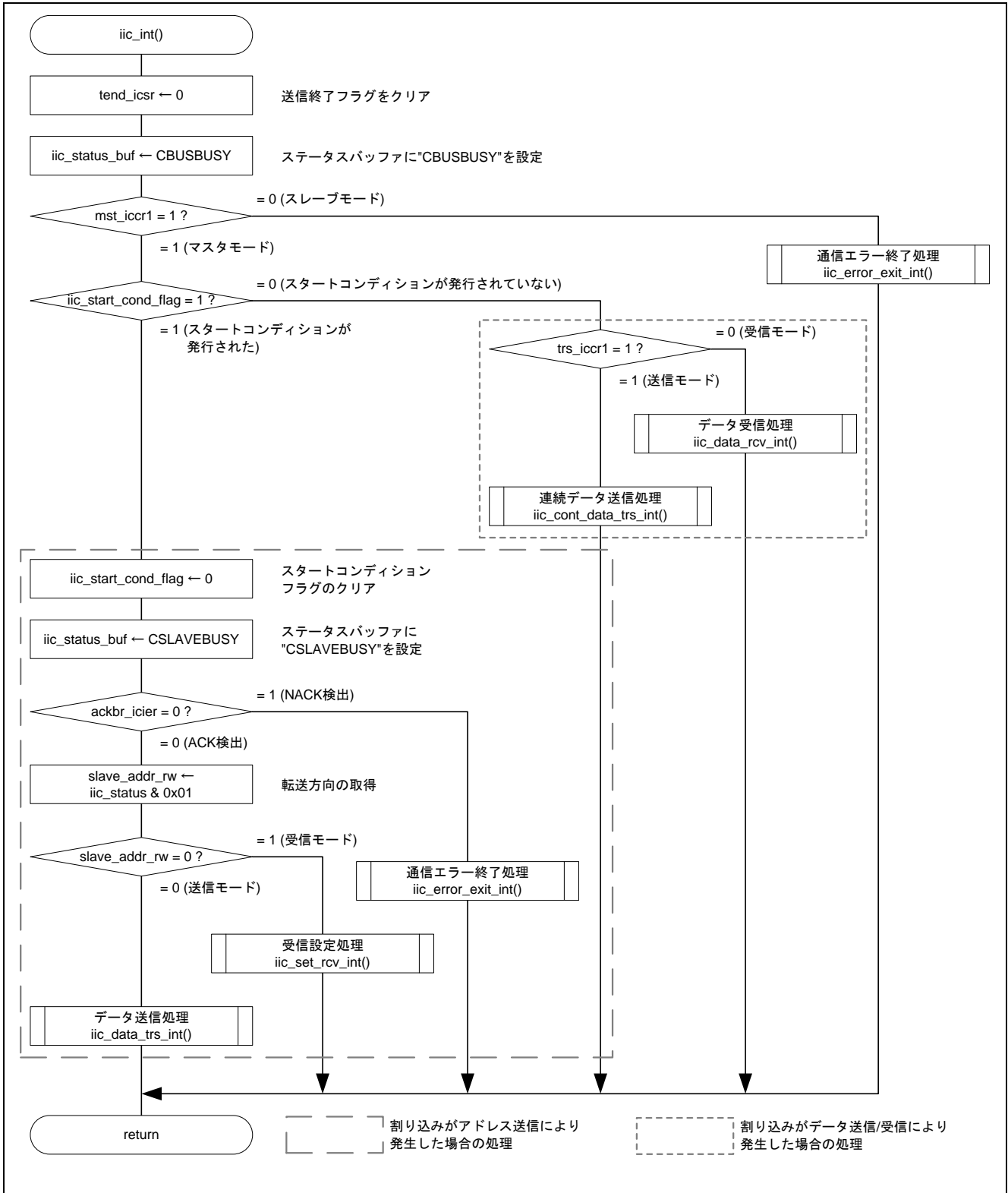


図 4.22 I²C バスインタフェース割り込み処理

4.21 データ送信処理

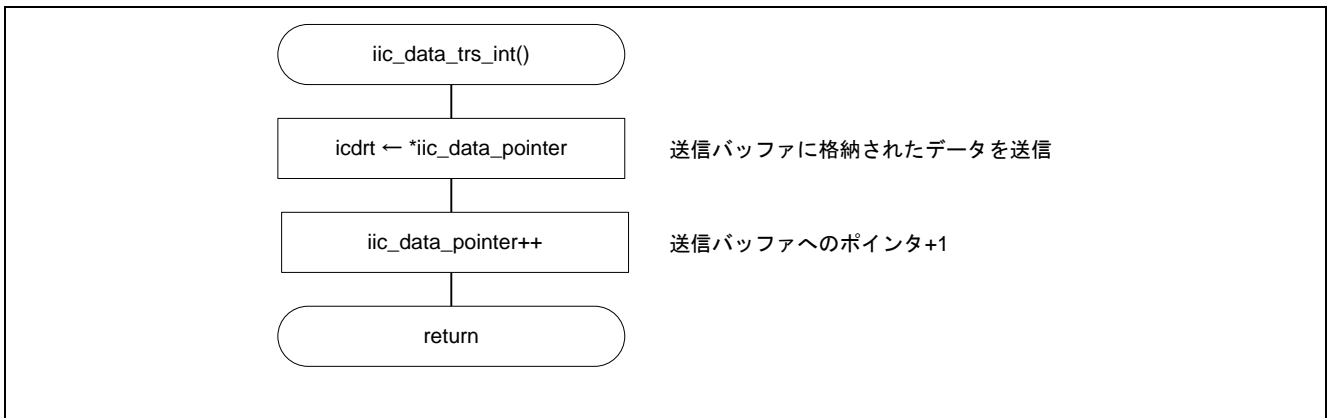


図 4.23 データ送信処理

4.22 連続データ送信処理

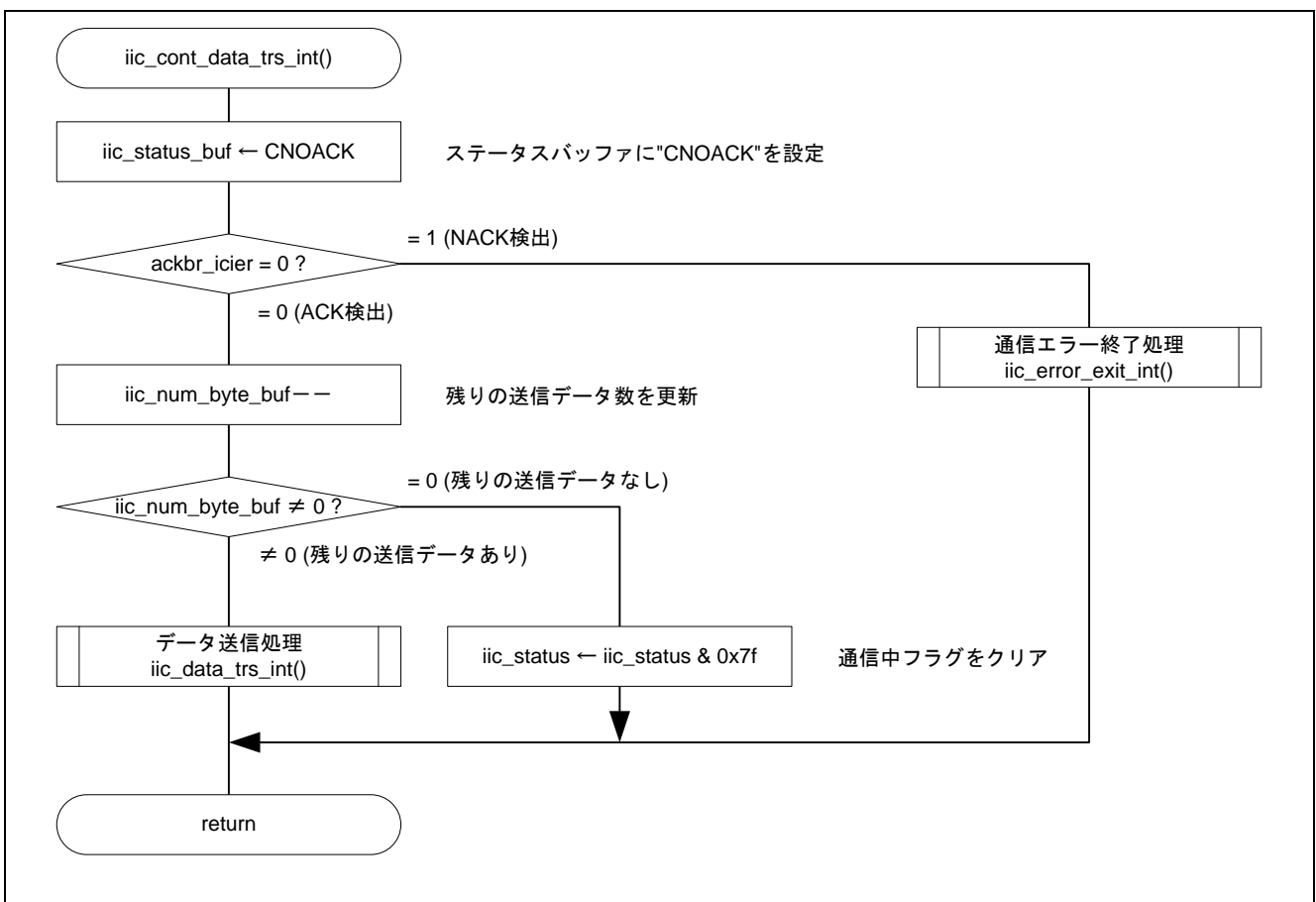


図 4.24 連続データ送信処理

4.23 受信設定処理

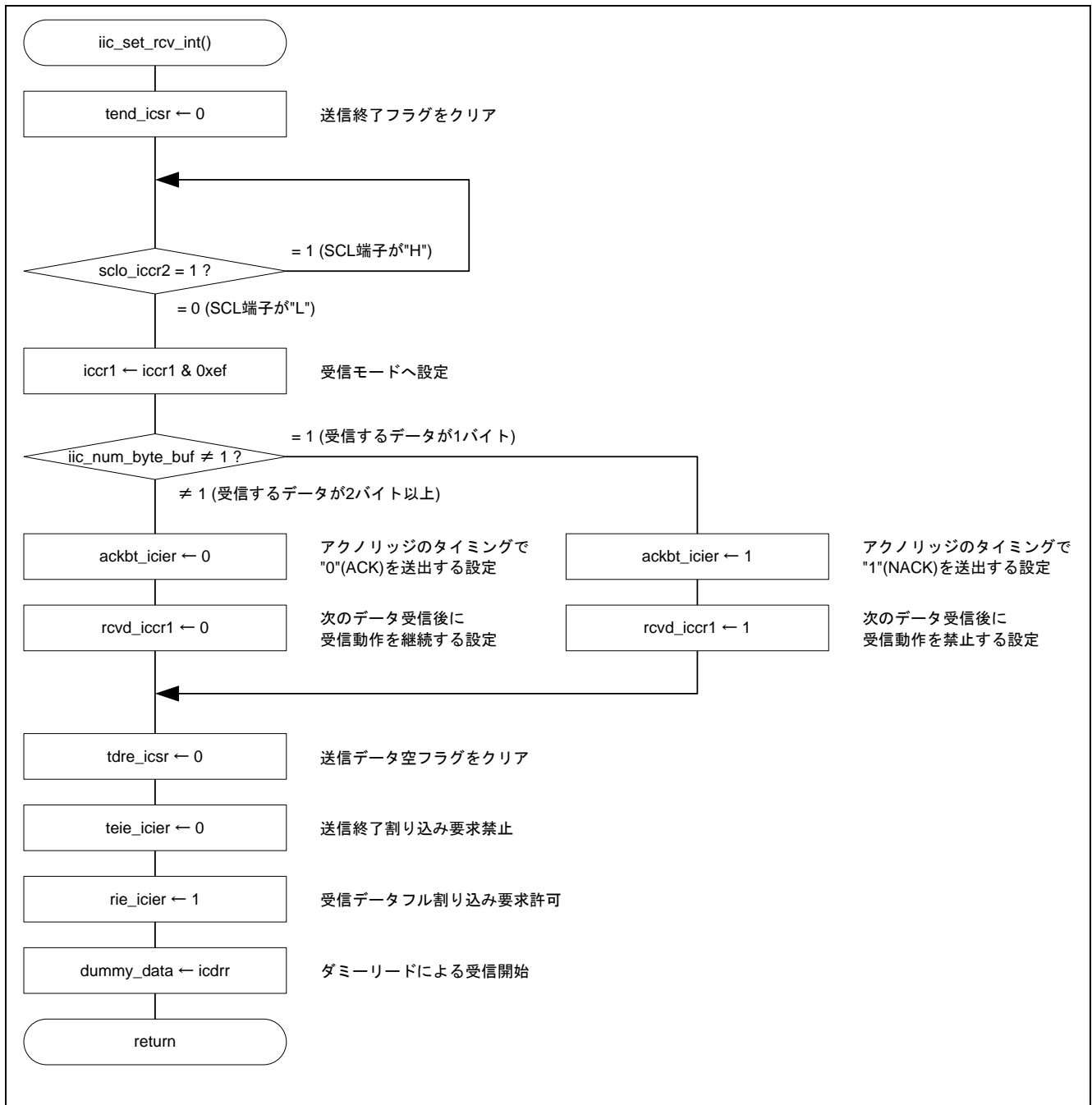


図 4.25 受信設定処理

4.24 データ受信処理

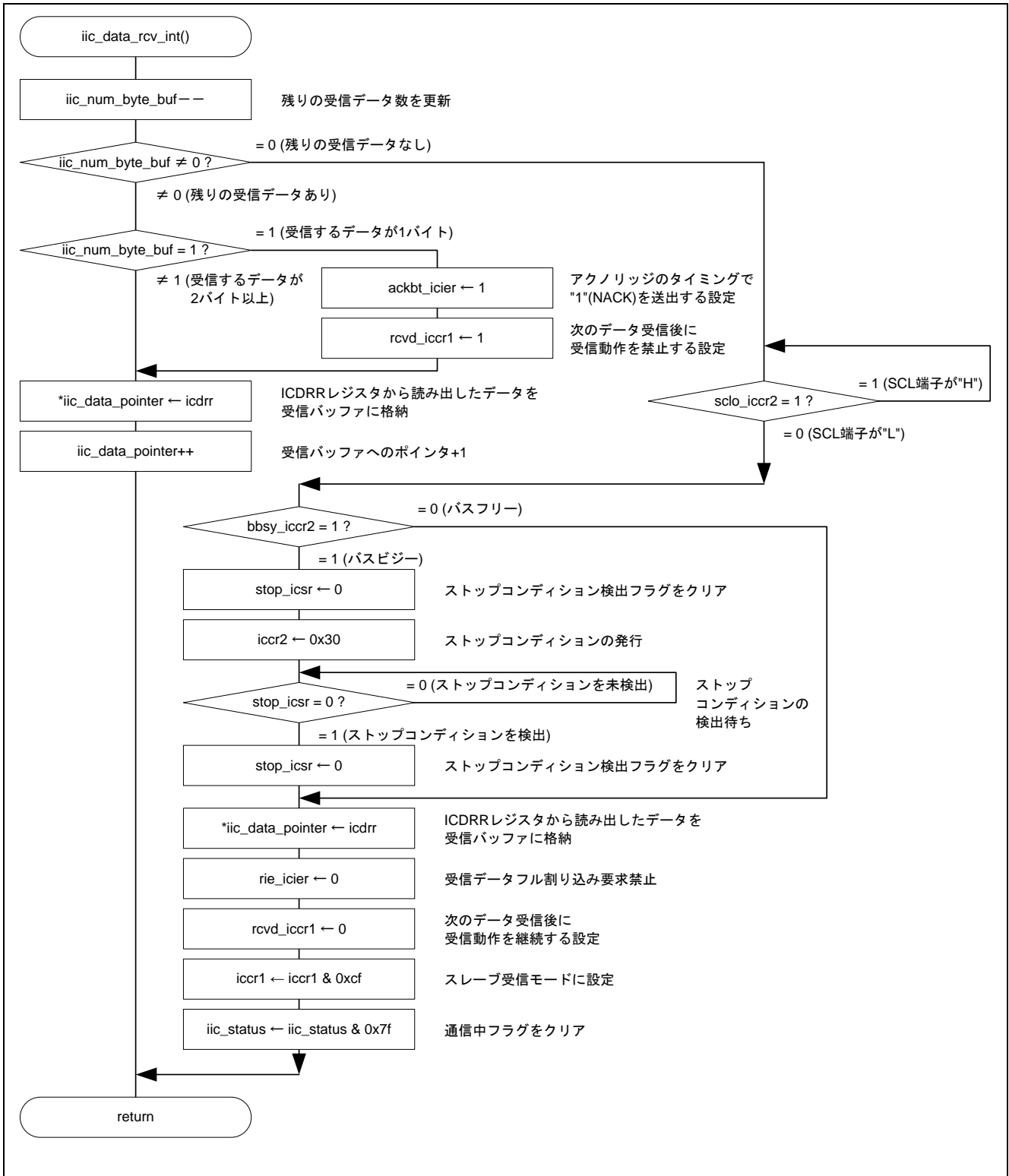


図 4.26 データ受信処理

4.25 通信エラー終了処理

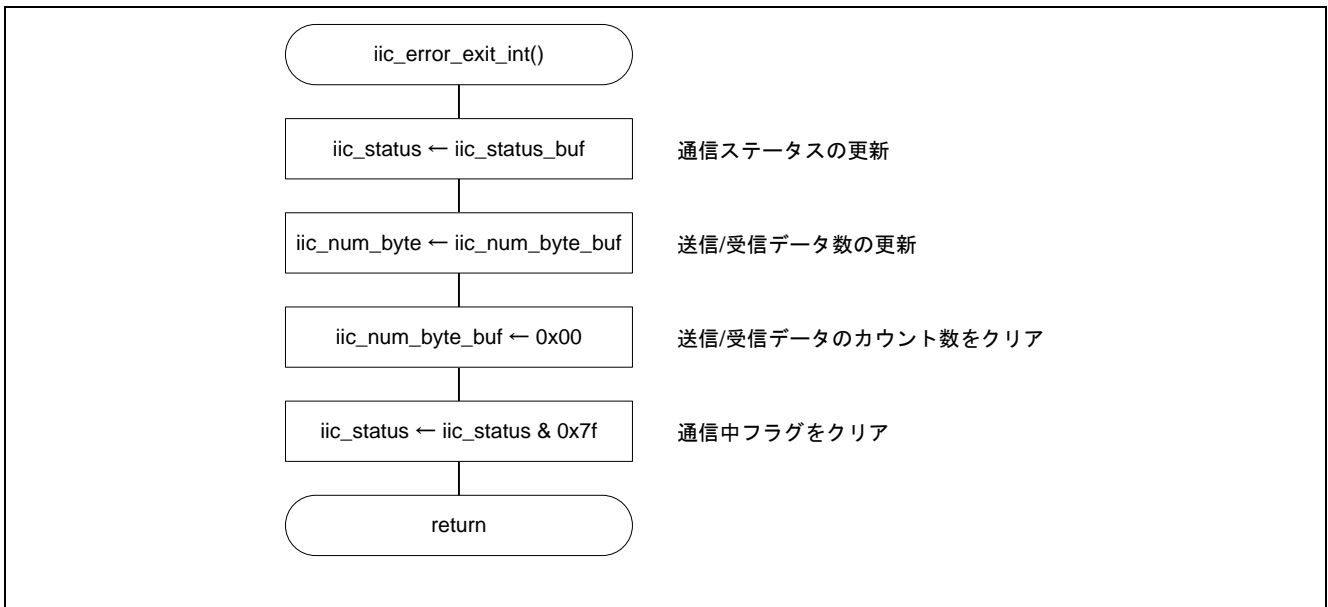


図 4.27 通信エラー終了処理

4.26 タイマ RA 割り込み処理

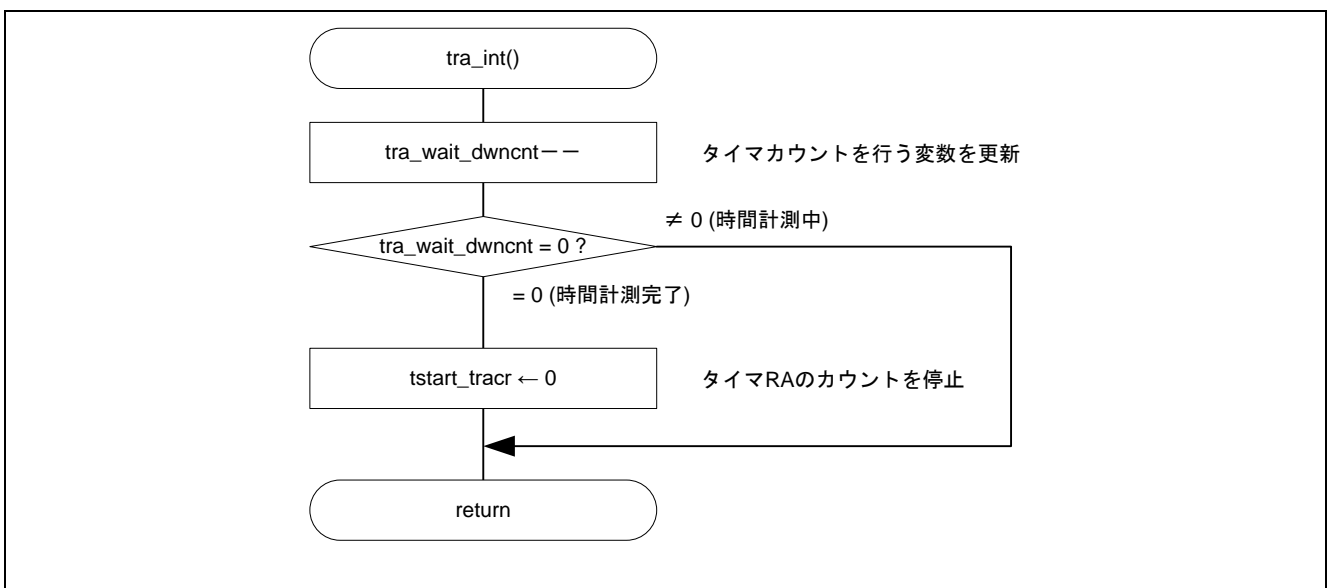


図 4.28 タイマ RA 割り込み処理

5. サンプルコード

サンプルコードは、ルネサスエレクトロニクスホームページから入手してください。

6. 参考ドキュメント

R8C/35C グループ ユーザーズマニュアル ハードウェア編 Rev.1.00

(最新版をルネサスエレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサスエレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

ルネサスエレクトロニクスホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017.05.11	-	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部ROM、レイアウトパターンの相違などにより、電氣的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が異なる製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれかに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、
金融端末基幹システム、各種安全制御装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することはできません。たとえ、意図しない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
 6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 10. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 12. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.3.0-1 2016.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>